Yannis Ioannidis Paolo Manghi Pasquale Pagano
(Editors)

Proceedings of the Second Workshop on Very Large Digital Libraries VLDL2009

A Workshop in conjunction with the European Conference on Digital Libraries 2009

Organized by the Institute of Information Science and Technology (ISTI) of the Italian National Research Council (CNR) of Pisa in cooperation with the DELOS Association
Corfu, Greece, 2nd of October 2009

Editors

Yannis Ioannidis
Department of Informatics,
National and Kapodistrian University of Athens,
Athens, Greece

Paolo Manghi
Istituto di Scienza e Tecnologie dell'Informazione (ISTI),
Consiglio Nazionale delle Ricerche,
Pisa, Italy

Pasquale Pagano
Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Consiglio Nazionale delle Ricerche, Pisa, Italy

Preface

The implementation of nowadays Digital Libraries is more demanding than in the past. Information consumers are facing with the need to access ever growing, heterogeneous, possibly federated Information Spaces while information providers are interested in satisfying such needs by sharing rich and organised views over their information deluge. Because of their fundamental role of information production and dissemination vehicle, Digital Libraries are also expected to provide information society with services that must be available 24/7 and guarantee the expected quality of service. This scenario leads to the development of Very Large Digital Libraries in terms of number of Information Objects and Collections to be made available, users to be served and potentially distributed resources needed to implement such systems. Such systems have to confront with new challenges in a context having scalability, interoperability and sustainability as focal points.

Interestingly, however, there is no clear and well-accepted definition of Very Large Digital Library (VLDL). Some, in an attempt to give one, blur the separation between Very Large Databases (VLDBs) and VLDLs and regard the latter as VLDBs storing Digital Library content. Since in databases the adjective Very large strictly refers to size of content, the implication is that, similarly, VLDLs ought to be DLs storing digital content beyond a given storage threshold. Despite being intuitively correct, this explanation does not fully satisfy DL practitioners. Indeed, DL content design paradigms cannot be conceptually separated by the relative functionalities as it happens for DBs. DLs are of use to peculiar user communities whose functionality needs, best practices and behaviour are well-accepted DL systems requirements. As captured by the DELOS reference model for DLs, user management, content management, functionality management, and policies are equally important in the definition of a DL. Accordingly, as demonstrated by real DL system experiences, VLDLs should be described as DLs featuring very large features in one or more of such aspects.

Since its first edition (ref. VLDL2008 Workshop Report, published in SIGMOD Records 2009), the goal of this workshop was to provide researchers, practitioners and application developers with a forum fostering a constructive exchange among all key actors in the field of Very Large Digital Libraries. Its long-term and ambitious mission is to discuss and delineate the foundations of VLDLs as a research field on its own, with well-defined areas, problems, solutions and open questions.

We would like to thank all the people who have directly or indirectly contributed to the event. Among these our colleagues at ISTI-CNR, Donatella Castelli, Leonardo Candela, and Costantino Thanos for their research inspiration, and Vittore Casarosa and Francesca Borri for the logistic support. Finally, our sincere gratitude goes to the members of the program committee, who sacrificed part of their precious time for the success of this workshop, to the chairs of the sessions, who offered their expertise to coordinate and harmonize the sequence of presentations, and of course to all authors, whose passion and ideas were the real fuel of VLDL 2009.

Yannis Ioannidis, Paolo Manghi and Pasquale Pagano
Organizers and Editors of the second VLDL workshop

Program Committee - Reviewers

Stavros Christodoulakis
Technical University of Crete (MUSIC/TUC),
Greece

Stefan Gradman
Institut fr Bibliotheks und Informationswissenschaft,
Humboldt-Universitt zu Berlin,
Germany

Kat Hagedorn
OAIster Project,
University of Michigan Digital Library Production Service,
USA

Dean B. Krafft
National Science Digital Library Project,
Cornell Information Science,
USA

Yosi Mass
IBM Research Division,
Haifa Research Laboratory,
University Campus, Haifa, Israel

Peter Wittenburg
Max-Planck-Institute for Psycholinguistics,
The Netherlands

Supporting Institutions

VLDL 2009 benfited from the support of the following organizations:

- DELOS Association
- Institute of Information Science and Technology of the Italian National Research Council (ISTI-CNR), Pisa, Italy
- Organization of the European Conference on Digital Libraries 2009

Table of Content

Invited talks

Building Europeana v1.0: towards a Large-Scale Content Ingestion
Julie Verleyen

SAPIR: towards Large Scale Multimedia Content Search
Maristella Agosti and Fausto Rabitti

VLDL Systems

Utility-based High Performance Digital Library Systems
Hussein Suleman

MultiMatch: Multiple Access to Cultural Heritage
Giuseppe Amato, Franca Debole, Carol Peters, Pasquale Savino

Integrating Multi-Dimensional Information Spaces
Kostas Saidis and Alex Delis

Data Management in VLDLs

Improving Similarity Search in Face-Images Data
Pedro Chambel, Fernanda Barbosa

Improving Query Results with Automatic Duplicate Detection
Irina Astrova

Enabling Content-Based Image Retrieval in Very Large Digital Libraries
Paolo Bolettieri, Andrea Esuli, Fabrizio Falchi, Claudio Lucchese, Raffaele Perego, and Fausto Rabitti

Functionality for VLDLs

Semantic Journal Mapping for Search Visualization in a Large Scale Article Digital Library
Glen Newton, and Alison Callahan and Michel Dumontier

Exploiting Individual Users and User Groups Interaction Features: Methodology and Infrastructure Design
Emanuele Di Buccio and Massimo Melucci

Maintaining Object Authenticity in Very Large Ddigital Libraries
Tobias Blanke, Stephen Grace, Mark Hedges, Gareth Knight, and Shrija Rajbhandari

Building Europeana v1.0: towards a Large-Scale Content Ingestion

Julie Verleyen

Europeana Office,
Koninklijke Bibliotheek,
National Library of the Netherlands

Europeana version 1.0 is an EU project that will develop a fully operational Europeana website. This will allow users to access over 10 million digital items from Europe's rich cultural and scientific heritage by 2010.

This - already important - challenge is only the top of the iceberg of content that will continue to flow *to* and *from* Europeana after the release of the first operational version of the service.

Meeting this challenge implies the setting up of a system that supports a large-scale workflow: at one end of the chain there are processes that smoothly ingest data submitted by Content Providers and Aggregators, while (web) services at the other end interact with components giving access to the data. We present here the preparation of such system and in particularly its standard-based approach to the functional modelling of the workflow, using the Open Archival Information System (OAIS) Reference Model with a focus on what happens at the interface between the data submitters and Europeana, the receiver. Update, traceability, duplication management, conversion: Europeana addresses these functional needs at the scale of hundreds of institutions and millions of items.

We also illustrate that besides the technical solutions the organisational aspects become more and more crucial when everything increases (number and size of data sets, number of parties involved, multiplicity of sources). A sound network strategy, clear agreements, rules and workflows are to be carefully established next to performing tools for partner and data management system, tracking of information, data transfer and storage facilities.

SAPIR: Towards Large Scale Multimedia Content Search

Maristella Agosti[1] and Fausto Rabitti[2]

[1] Dipartimento di Ingegneria dell'Informazione,
University of Padua,
Padua, Italy

[2]Istituto di Scienza e Tecnologie dell'Informazione (ISTI),
Consiglio Nazionale delle Ricerche (CNR),
Pisa, Italy

Existing web search technologies are limited to text-based search, yet still 99% of the information on the web consists of audio-visual content that is searchable only by associated metadata and not by its actual content. This amazing restriction has raised the question of how search technologies can tap into the potential reservoir of information. SAPIR (*Search on Audio-visual content using Peer-to-peer Information Retrieval*) has developed cutting-edge technology to break the barriers and enable search engines to *search large-scale, audio-visual information*, by content. SAPIR's distributed P2P technology has proven to be able to effectively deal with the fundamental scalability issue.

The main aim of SAPIR has been to develop theories and technologies for next-generation search techniques that would effectively and efficiently deliver relevant information in the presence of exponentially growing (i.e., dynamic) volumes of distributed multimedia data. Fundamental to our approach is the development of scalable solutions that address the requirements of future generations of massively distributed data produced in a variety of applications and available on the Web.

Utility-based High Performance Digital Library Systems

Hussein Suleman

Department of Computer Science, University of Cape Town,
Private Bag X3, Rondebosch, 7701, South Africa
hussein@cs.uct.ac.za

Abstract. Many practical digital library systems have had to deal with scalability of data collections and/or service provision. Early attempts at enabling this scalability focused on data/services closely coupled with or tightly integrated with various high performance computing platforms. This inevitably resulted in compromises and very specific solutions. This paper presents an analysis of current high performance systems and motivates for why utility computing can subsume existing models and better meet the needs of generic scalable digital library systems.

1 Introduction

Cloud computing is the current but ill-defined buzzword in computing for the use of abstract online resources [15] - utility computing is a specific subset of cloud computing whereby virtual computing and storage resources are acquired (or released) on-demand without the consumer of the resource knowing their physical location [10]. The most common scenario is the need for a large numbers of computers for a short period of time, for example to index a large ingested collection. Utility computing is arguably more cost-effective than other forms of high performance computing (HPC) because of the efficiency of shared resources for which the management is essentially outsourced.

Designers of digital library systems (DLSes) have long grappled with the problem of how to construct systems that meet multiple criteria, including elements of scalability and preservation of data [7]. As HPC technologies have emerged and matured, DLSes were designed to exploit them. These include symmetric multiprocessors, clusters and grids. Utility computing, a relatively newer paradigm, is arguably a better fit because it provides an ideal compromise among features to enable high performance digital libraries that can be provisioned as needed (i.e., on-demand).

The rest of this paper provides an overview of current high performance computing technology and how these are utilised in the provision of digital library services, and then how an alternative digital library system can be designed based on utility computing. The focus is on service provision rather than data scalability as service provision must be built over utility computing platforms while scalable data storage is often a basic service.

2 High Performance Computing

High performance computing is about the construction of hardware and software systems to process a computationally-intensive job. This often is distinguished from high throughput computing, which is about systems to process a large number of jobs. As these are not completely disjoint, the former term has become the norm to describe both forms of computing.

Many forms of HPC have been adopted in computer systems, including: cluster computing, grid computing, cloud/utility computing, edge computing, symmetric multiprocessing, multicore computing, general-purpose graphics processing units, cell processors, field programmable gate arrays, server farms and volunteer computing. Each of these is discussed briefly below.

2.1 CPU Models

The earliest form of scalability in online systems was effected using server farms. These are individual machines, each able to respond to requests from the Web, with possibly a shared database backend or independent replicated data stores (if the site is mostly read-only). Server farms are effective when the majority of processing does not require database updates - in this case the round-robin IP address resolution from the DNS system automatically distributes the load evenly among the servers. DSpace implementers have recommended the use of such a technique for some degree of scalability because it is supported by the Web server and the architecture of DSpace [11].

Modern multicore CPUs contain multiple processing units in a single microchip. Each core is capable of logic and arithmetic operations, with a single shared memory for all cores. On-chip caches can be shared in a multicore CPU to allow for a high degree of integration. Multicore machines are supported by operating systems at the process/thread level - the operating system is able to allocate different processes to run on different cores simultaneously. Thus, an application that is appropriately split into sub-tasks will automatically take advantage of the additional cores. Most desktop computers and servers use multicore CPUs, although most programming models are still serial.

Symmetric multiprocessing refers to computers that contain multiple independent CPUs on separate microchips. These also are managed by modern operating systems at the process level and behave very similarly to multi-core systems. They are not, however, as efficient for inter-process communication because this is between-chip rather then within-chip. SMP systems were popular in early supercomputers and were the focus of early attempts to achieve scalability in digital library systems [4].

Mosix [5] is an operating system tool that binds a group of networked machines into a single virtual operating system. The programmer or end user sees a group of machines as a single machine and Mosix handles the distribution of processes and inter-process communication. Mosix has the advantages of SMP from an end-user perspective, but has a much slower interconnecting network between processors. Mosix, SMP and multicore systems were used in experiments with

parallelising metadata harvesting - and it was shown that the same applications would work on all architectures with varying performance [19].

Cell processors [14] contain a main controlling core but many sub-cores for execution of data-parallel tasks. Cell processors were made popular by their incorporation into the Playstation 3 video games consoles for graphics processing.

General-purpose Graphics Processing Units (GPGPUs) [17] exploit the increasing computational power of microchips on commodity graphics cards, which are able to perform some operations in parallel, thus resulting in much faster processing than traditional CPUs. As such, they are useful for a large number of computation-intensive tasks. Tools for programming GPUs are still in their infancy and high end GPUs are not yet considered commodity components.

Field-Programmable Gate Arrays (FPGAs) [9] are microchips that can perform specific tasks in hardware which is reconfigured at runtime. This allows for maximum performance, although there are limitations in the size of FPGAs and there is still a need for a CPU for general purpose processing. FPGAs are most suited to highly parallel computation-intensive tasks.

There are currently no well-known digital library applications that explicitly use cell processors, GPGPUs or FPGAs.

2.2 Machine/Storage Models

Cluster computing is when a group of co-located standalone computers - usually commodity machines - are interconnected into a LAN using a high speed network fabric. Each machine on the cluster runs an independent task or set of tasks, with coordination taking place over the high speed network. Clusters are suited to problems which can be decomposed into separate processes with some communication. The amount of communication is closely related to the speed of the interconnect, so general purpose clusters tend to use fairly high-end interconnects such as Gigabit Ethernet. Clusters have been used in scientific computing for space-partitioned discrete event simulations. Most commercial search engines such as Google use a cluster architecture for their indexing and querying operations because of the high memory bandwidth required for efficient indexing (whether for search or browse indices) [6].

Grid computing refers to a coordinated sharing of separately-owned resources in a wide area network, typically overlaid on the Internet [13]. Grid computing encompasses a suite of technologies to control access to resources, support data transfers and manage processes on remote systems. As the computational and storage resources are potentially far apart, the network interconnect is not as controlled and speeds are typically far lower than with clusters. As a result, grids are suitable where there is little communication among processes and where the data to be transferred is small relative to the amount of computation to be performed. In the digital library community, grids have been used for both computation and storage. Storage Resource Broker and iRODS [18] are used to create data (storage) grids over hetergeneous resources. DILIGENT [8] is a digital library management system that used a grid infrastructure to interconnect services distributed over a wide area.

Edge computing is the use of computers at the edges of the network to perform computation, instead of machines at the core. While this is not rigorously defined, there are some examples in practice that qualify as edge computing. Content distribution networks are one example, where content and sometimes services are distributed to sites closest to the end-user. This makes the service provision scalable by reducing the network distance from user to service. Another example is the use of the user's desktop machine to perform some of the computation instead of all computation happening on servers. Asynchronous Javascript and XML (AJAX) Web applications are able to scale to a far greater degree because much of the computation occurs on the clients' computers. In the Bleek and Lloyd project, a prototype search engine was created in AJAX to demonstrate how ranked retrieval could be performed on the client using local data, thus eliminating the need for network interaction and resources on the server [20].

Volunteer computing is the use of idle desktop computers to perform computation for which it is not possible to secure dedicated high performance systems. SETI@Home popularised this form of distributed computing for the large-scale analysis of astronomical data and the technology has since evolved into the general purpose BOINC framework for sharing CPU cycles [3]. Volunteer computing has great potential for scalability but sometimes limited applicability because of the uncertain widely distributed resources and unclear trust relationships. No known digital library systems are using volunteer computing.

Figure 1 illustrates key differences among these machine models, arranged from centralised to distributed. The dotted line cutting across the categories indicates how utility computing compares against the other machine models.

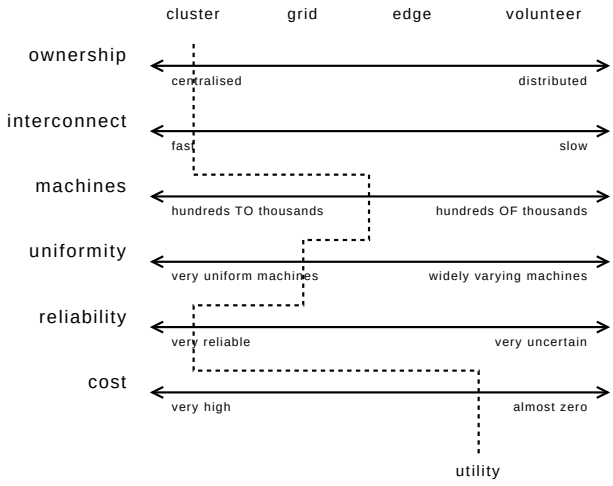


Fig. 1. HPC Machine configurations

2.3 Utility Computing

Utility computing has emerged as a generalisation of most of the existing HPC models, where the exact nature of the technology is not visible to the end-user or the programmers. Instead all interaction is via an API and the user may assume that the machines are interconnected with a high-speed network, with localised access to data.

Two core services are provided in utility clouds: virtual machines and virtual data stores. Virtual machines are machines that can be created on demand, often programmatically, and used remotely. Amazon's Elastic Compute Cloud (EC2) [1] is a popular example of this service. Virtual data stores allow the storage of a virtually unrestricted amount of information, with a uniform addressing scheme. Amazon's Simple Storage System (S3) [2] is a popular example of this service.

The provisioning of utility computing can be done either remotely as an outsourced operation, or internally using similar APIs. Eucalyptus [16] is a software suite that emulates the APIs created by Amazon, allowing an organisation to provision its own utility computing services, while still retaining the benefit of a clean separation between services and provisioning.

As shown in Figure 1, utility computing has the benefits of clusters in terms of ownership of resources being centralised, high speed interconnects, a reasonable level of homogeneity of machines and a high degree of reliability. In addition, there are generally many more machines available than in a typical cluster (grid-like), while the cost of processing is much smaller since there is no upfront and continuous hardware acquisition cost (volunteer computing-like).

3 Designing Utility Computing HPDLs

While utility computing systems may offer a single scalable data store, there is currently no single scalable machine architecture. Applications therefore must be designed to make optimal use of virtual machines, possibly adopting the various architectures of pre-utility systems [21]. Figure 2 illustrates 4 possible architectures to incorporate scalability within the utility cloud.

The Proxy architecture has a manager node act as a proxy for all external connections to service nodes. This is similar to a cluster.

The Redirector architecture has the manager node act as a lookup table for service nodes, with the external client making direct connections into the cloud thereafter. This is similar to a grid.

In the Round Robin architecture the client uses the DNS system (or a similar resolution service) to obtain the address of the next service node to use. DNS typically rotates IP addresses to balance the load but the manager must still update the DNS tables based on the number of service nodes available. This is similar to a server farm.

In the Client-side model, the manager sends a list of nodes to the client upon request and the client uses this list to rotate requests to service nodes. This is similar to an edge computing model. This model ought to have the highest level of scalability and reliability.

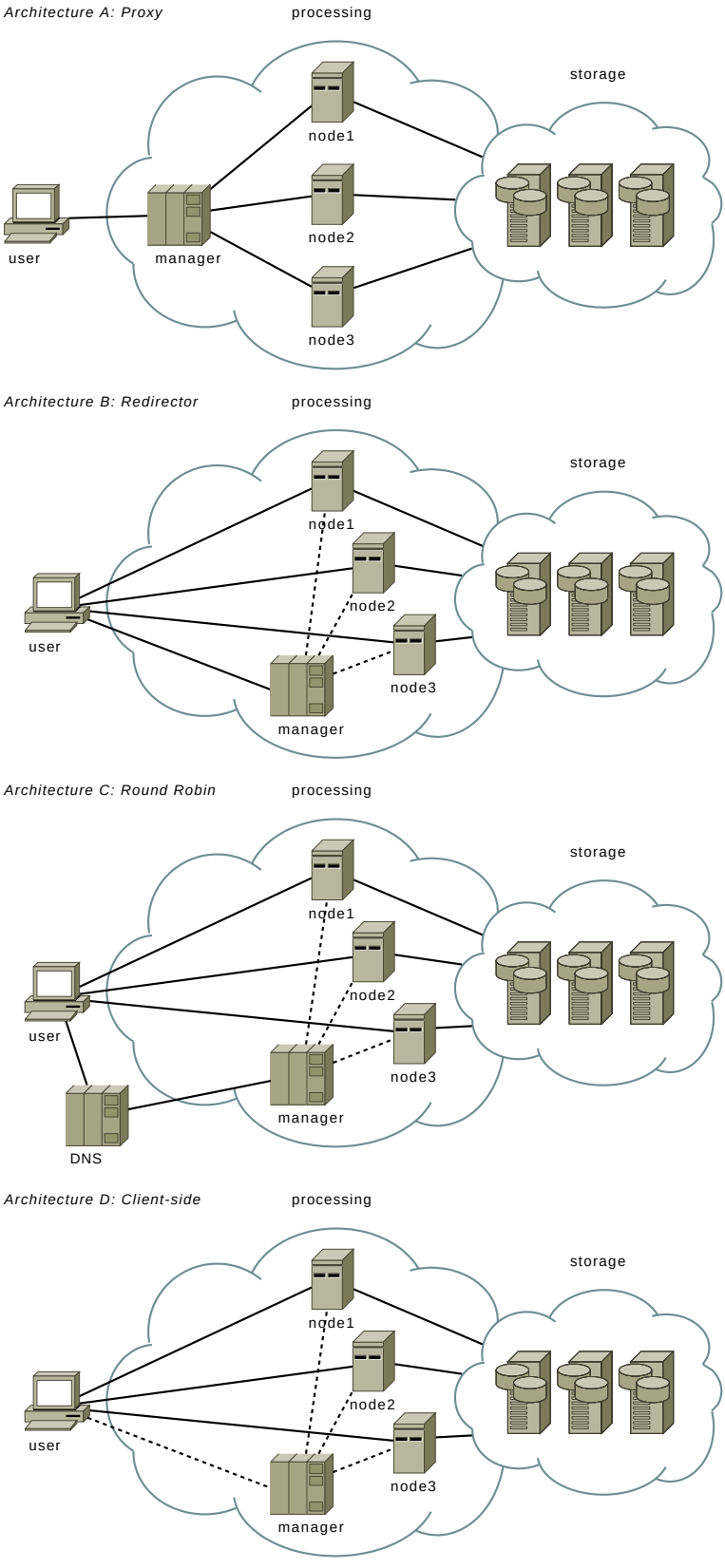


Fig. 2. Cloud Computing machine configurations

Thus, within a utility cloud, digital library services can be designed to emulate most existing HPC architectures - given that different DL services are most efficient on different architectures, utility computing offers the flexibility of multiple architectural models.

4 Conclusions and Future Work

Utility computing shows promise to provide computing facilities with the best features of multiple HPC technologies and substantial flexibility for implementing scalable systems. The system architectures presented here are currently being developed into prototype digital repository systems to experimentally verify scalability for popular services.

There is, however, still a need to balance scalability with preservation. The rapid advances in HPC technology has meant that no one option has been preferred by the DL community. Thus, there are no standard approaches to scalability and this makes preservation even more difficult. The Duraspace project is attempting to build more preservable cloud stores by replicating data across multiple cloud platforms, but this is still a work in progress [12].

Utility computing narrows the gap between HPC and on-demand computing by making resources available as required. Future work will focus on how to make use of those resources most cost-efficiently, without human intervention or with minimal intervention. This is vital for small collections to scale up in size or services without a redesign of the digital library system.

Finally, utility computing is not common in most developing countries because the Internet connection into the utility cloud is not fast enough. More work is needed on the localisation of utility clouds to ensure equal access to the technology.

5 Acknowledgements

This project is made possible by funding from University of Cape Town and NRF.

References

1. Amazon.com (2009). Amazon Elastic Compute Cloud. Website http://aws.amazon.com/ec2/
2. Amazon.com (2009). Amazon Simple Storage Service (Amazon S3). Website http://aws.amazon.com/s3/
3. Anderson, D. P. (2004). BOINC: A system for public-resource computing and storage, in Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, 8 November 2008, ACM Press.
4. Andresen, D., Yang, T., Egecioglu, O., Ibarra, O. H., and Smith, T. R. (1996). Scalability Issues for High Performance Digital Libraries on the World Wide Web, ADL, pp. 139, Third International Forum on Research and Technology Advances in Digital Libraries (ADL '96), IEEE Computer Society.

5. Bar, M. (2003). openMosix, a Linux Kernel Extension for Single System Image Clustering. in Proceedings of Linux Kongress: 10th International Linux System Technology Conference, 15-16 October, Saarbrücken, Germany.
6. Barroso, L. A., Dean, J., and Hölzle, U. (2003). Web Search for a Planet: The Google Cluster Architecture, IEEE Micro, March-April 2003, IEEE Computer Society.
7. Borgman, C. L. (2002). Challenges in Building Digital Libraries for the 21st Century, in Proceedings of Digital Libraries: People, knowledge, and technology : 5th International Conference on Asian Digital Libraries, ICADL 2002, Singapore, 11-14 December 2002, Springer.
8. Candela, L., Castelli, D., Pagano, P., and Simi, M. (2005). Moving Digital Library Service Systems to the Grid, Lecture Notes in Computer Science, Volume 3664, Springer.
9. Compton, K., and Hauck, S. (2000). An Introduction to Reconfigurable Computing, IEEE Computer, April 2000, IEEE Computer Society.
10. Danielson, K. (2009). Distinguishing Cloud Computing from Utility Computing, eBizQ, 26 March 2008. Available http://www.ebizq.net/blogs/saasweek/2008/03/distinguishing_cloud_computing/
11. DSpace.org (2009). Clustering, DSpace Wiki. Available http://wiki.dspace.org/index.php/Clustering
12. Duraspace (2009). Duracloud. Website http://www.duraspace.org/duracloud.html
13. Foster, I., and Kesselman, C. (2004). The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann.
14. Gschwind, M., Hofstee, H. P., Flachs, B., Hopkins, M., Watanabe, Y., and Yamazaki, T. (2006). Synergistic Processing in Cell's Multicore Architecture, IEEE Micro, March-Apruil 2006, IEEE Computer Society.
15. Knorr, E., and Gruman, G. (2008). What cloud computing really means, Infoworld, 7 April 2008. Available http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031
16. Nurmi, D., Wolski, R., and Grzegorczyk, C. (2008). Eucalyptus : A Technical Report on an Elastic Utility Computing Archietcture Linking Your Programs to Useful Systems, Computer Science Technical Report Number 2008-10, UCSB. Available http://www.cs.ucsb.edu/research/tech_reports/reports/2008-10.pdf
17. Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., and Purcell, T. J. (2007). A Survey of General-Purpose Computation on Graphics Hardware, Computer Graphics Forum, 26(1), pp. 80-113, Blackwell-Synergy.
18. Rajasekar, A., Wan, M., Moore, R., and Schroeder, W. (2006). A prototype rule-based distributed data management system, in Proceedings of HPDC Workshop on Next-Generation Distributed Data Management, 19-23 June 2006, Paris.
19. Suleman, H. (2006). Parallelising Harvesting, in Proceedings of 9th International Conference on Asian Digital Libraries, ICADL 2006, Kyoto, 27-30 November 2006, Springer.
20. Suleman, H. (2007). in-Browser Digital Library Services, in Kovacs, Laszlo, Norbert Fuhr and Carlo Meghini (eds): Proceedings of Research and Advanced Technology for Digital Libraries, 11th European Conference (ECDL 2007), pp. 462-465, 16-19 September, Budapest, Hungary. Available http://pubs.cs.uct.ac.za/archive/00000434/01/ecdl_2007_ajax.pdf
21. Wayner, P. (2008). Cloud versus cloud: A guided tour of Amazon, Google, AppNexus, and GoGrid, InfoWorld, 21 July 2008. Available http://www.infoworld.com/d/cloud-computing/ cloud-versus-cloud-guided-tour-amazon-google-appnexus-and-gogrid-122

MultiMatch: Multiple Access to Cultural Heritage

Giuseppe Amato, Franca Debole, Carol Peters, Pasquale Savino

Institute of Information Science and Technologies - CNR
Pisa, Italy
{franca.debole}@isti.cnr.it

Abstract. Our shared cultural heritage (CH) is an essential part of our European identity, transcending cultural and language barriers. The aim of the MultiMatch project is to enable users to explore and interact with online internet-accessible CH content, across media types and language boundaries, in ways that do justice to the multitude of existing perspectives. This has been achieved through the development of a search engine targeted for the access, organisation and personalized presentation of large and complex cultural heritage information. MultiMatch aims at heterogeneous digital object retrieval and presentation.

1 Introduction

Online Cultural Heritage (CH) content is being produced in many countries by organisations such as national libraries, museums, galleries and audiovisual archives. Additionally, there are increasing amounts of CH relevant content available more generally on the World Wide Web. While some of this material concerns national or regional content only of local interest, much material relates to items involving multiple nations and languages, for example concerning events in Europe or Asia. In order to gain a full understanding of such events, including details contained in different collections and exploring different cultural perspectives requires effective multilingual search technologies. The EU FP6 MultiMatch project is concerned with information access for multimedia and multilingual content for a range of European languages. MultiMatch tries to offer "complex object retrieval" through a combination of focused crawling, and semantic enrichment that exploits the vast amounts of metadata available in the cultural heritage domain. The MultiMatch search engine was developed with specialised search facilities for multilingual access to cultural heritage material in diverse media. The aim was to present the user with the "detailed picture" of complex CH objects to a very large part of the people interested on CH. The overall goal of the project was to build a fully operational system prototype, designed and refined according to the requirements of numerous and different user classes. Users can search information using their preferred language, searching for all types of digital objects, accessing only the sites that contain information potentially relevant to their request, retrieving mainly relevant items, and viewing the

query results in an organized structured fashion. Standard and ontology-based descriptions of content are used, thus providing an interoperable semantic framework for intelligent multimedia object delivery. Metadata automatically mapped from the original CH material and enriched with supplementary information was ingested onto this framework. At the moment the system has been demonstrated for the main languages of the cultural heritage institutions in the consortium: Dutch, Italian, Spanish, English and also German and Polish, but it is extendible to other languages.

2 Motivation

Europe's vast collections of unique and exciting cultural content are an important asset of our society. On the web, cultural heritage (CH) content is everywhere, in traditional environments such as libraries, museums, galleries and audiovisual archives, but also reviews in popular magazines and newspapers, in multiple languages and multiple media. CH objects on the web are no longer isolated objects, but situated, richly connected entities, equipped with very heterogeneous metadata, and with information from a broad spectrum of sources, some with authoritative views and some with highly personal views. The aim of the MultiMatch project is enabling users to explore and interact with online internet-accessible CH content, across media types and language boundaries, in ways that do justice to the multitude of existing perspectives. This has been achieved through the development of a search engine targeted for the access, organisation and personalized presentation of cultural heritage information. The main source of information stored in the MultiMatch prototype system is composed of cultural heritage objects obtained through crawling and indexing of material obtained from cultural heritage sites, web encyclopedias (e.g. Wikipedia), digital libraries of specific cultural heritage organizations, OAI compliant digital resources, and RSS feeds from cultural web sites. The cultural heritage search and navigation facilities envisaged by MultiMatch cater for these information needs by presenting users with a composite picture of complex CH objects. For instance, in reply to a user's request for information on Van Gogh, the MultiMatch engine can present information on Van Gogh from multiple museums around Europe, in multiple languages; it could complement this with pointers to Van Gogh's contemporaries, with links to exhibitions on Van Gogh, to reviews of these exhibitions, to blog entries by visitors to these exhibitions, and to background information taken from online resources or dedicated sites. The MultiMatch search engine has been developed with specialised search facilities for multilingual access to cultural heritage material in diverse media.

3 The System

The concepts underlying the system are shown in Figure 1. On the left-hand side of the figure, we show users querying the system in different languages for a range of information on the Dutch artist Vincent van Gogh, including critical

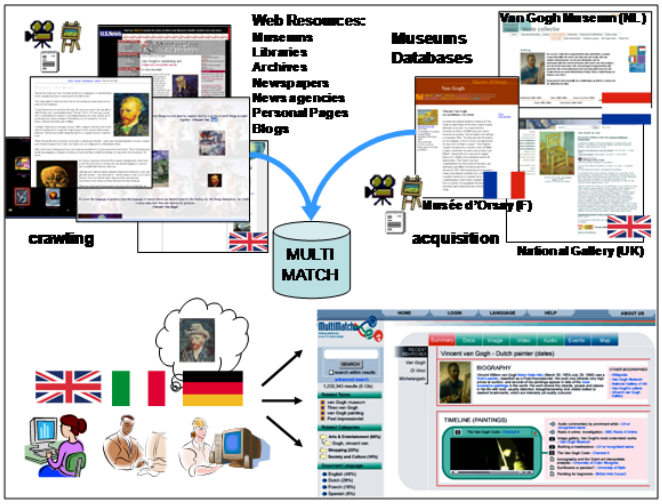


Fig. 1. Overview of the MultiMatch Integrated System.

analysis, biographies, details of exhibitions. The system displays the information retrieved in an integrated fashion, and in a format determined by the particular user profile. On the right-hand side, we show possible sources of this information and the ways in which it can be acquired. The project developed a system prototype that supports six languages: Dutch, English, Italian, German, Polish, and Spanish, and extendible to others. The MultiMatch search engine is able to:

- identify relevant material via an in-depth crawling of selected CH institutions, accepting and processing any semantic web encoding of the information retrieved;
- crawl the Internet to identify websites with CH information, locating relevant texts, images and videos, regardless of the source and target languages used to write the query and/or describe the results;
- automatically classify the results on the basis of a document's content, its metadata, its context, and on the occurrence of relevant CH concepts;
- automatically extract relevant information which will then be used to create cross-links between related material, such as the biography of an artist, exhibitions of his/her work, critical analysis, etc.;
- organise and further analyse the material crawled to serve focused queries generated from information needs formulated by the user;
- interact with the user to obtain a more specific definition of initial information requirements;
- the search results are organized in an integrated, user-friendly manner, allowing users to access and exploit the information retrieved regardless of language barriers.

The achievement of these objectives implied a significant research effort in the general field of multilingual/multimedia information access and retrieval, and in particular in the following areas:

Focussed search engine In its simplest form, a vertical search engine, i.e., one that caters for domain-specific searches, simply filters a subset of the web believed to be relevant to a topic. In a more useful form, a vertical search engine is able to extract information from web pages, allowing for more sophisticated query interfaces and presentation of results adapted to the task. MultiMatch aimed at taking a significant leap forward from today's vertical search engines, by offering "complex object retrieval" through a combination of focused crawling and semantic enrichment that exploits the vast amounts of metadata available in the cultural heritage domain, presenting both certified and non-certified information together (while clearly distinguishing one from the other). The MultiMatch project fits into the category of advanced, domain-specific search engines, with some salient features: i) it is a search engine that combines automatic classification and extraction techniques with semantic web compliant encoding standards; ii) it considers complex user profiles and search scenarios; iii) it is able to search across language boundaries and across different media.

Multilingual/multimedia indexing Instead of returning documents in isolation, MultiMatch provides complex search results that put documents of various media types into context. For the indexing-end of MultiMatch, complex object retrieval generates special challenges. First, documents of various media types (text, audio, image, video, or mixed-content) and accompanying metadata are indexed. Existing generic standards such as MPEG-7 cater for such data models by incorporating multimedia content and metadata in a single semi-structured document. The indexing strategies used also recognise and cater for multilingual content. Particular Cultural Heritage relevant Internet domains or subdomains were spidered using a state-of-the-art crawler [1] [2] and, in parallel, where supported by the CH institution, the engine interfaces with information sources using open standards [3]. CH information was also gathered from the Web at large, employing existing focused crawling techniques specifically targeting cultural heritage information [4],[5].

Information extraction and classification MultiMatch allows users to interpret the wealth of CH information by presenting objects not as isolated individual items, but as situated, richly connected entities. A range of classifications, as well as various links to reviews, experience reports, and general background knowledge, are provided. Documents are classified on the basis of diverse dimensions, such as topical, geographical, and temporal. MultiMatch uses large scale information extraction from documents to identify entities and their relations in large Web corpora [6] [7].

Multilingual/multimedia information retrieval For many years information retrieval research concentrated primarily on English language text documents.

However, recent years have seen a significant increase in research activity extension to information retrieval techniques for multimedia and multilingual document collections. Unfortunately, so far, there has been little transfer of research advances to real world applications. MultiMatch aimed at bridging this gap. Multimedia data can be classified according to its constituent media streams: audio, visual and textual. Research in audio retrieval has largely been concentrated in Spoken Document Retrieval (SDR), where the key challenge is accurate automatic content recognition. Research in Visual Information Retrieval (VIR) for images and video data streams has similarly been underway for over 10 years. Problems of VIR relate to both recognition of visual content and the definition of visual content for IR. MultiLingual Information Retrieval (MLIR) has also become an established area of research in recent years. MLIR focuses on the problem of using a request in one language to retrieve documents from a collection in multiple different languages. MultiMatch developed components for both document and query translation and procedures for matching one against the other. Much effort was dedicated to the building of domain-specific multilingual resources catering for the terminology adopted in the CH domain [8].

User-centred interaction Although there has been huge progress, content-based information retrieval (e.g. video and image retrieval by visual content) still faces significant barriers when attempting to create truly effective and comprehensive retrieval with respect to the user's needs. Users look mainly for concepts (e.g. individuals, facts, places) and far less for features (e.g. mountains, sunset, clouds). A "semantic gap" exists: human beings intrinsically interpret images depending on a subjective viewpoint while computers remain at the most objective and elementary level. To bridge the semantic gap, human intervention is still needed to add high-level features (i.e. metadata) [9]. However, recent advances in the areas of information retrieval and information extraction make it possible to automatically associate concepts to objects when text is available. The MultiMatch user interface integrates automatic techniques for low level feature extraction and automatic concept classification. Structures for browsing are created, allowing users to explore content or search results following multiple facets. A key research problem for MultiMatch was enabling the user to adequately formulate their query using the language of their choice and specify both low-level and high-level multimedia feature [10].

4 Data and Evaluation

The MultiMatch prototype included a wealth of documents derived from multiple sources, annotated by using different formats and schemas, and composed of different types of data, and different languages. During the project lifetime we selected, analysed, harmonized and mapped partners collections and external cultural heritage data. The following Figure (2) describes some of the data ingested on the MM engine from a quantitative (and global) point of view. The prototype system was evaluated in order to measure its quality with respect to

Origin	Sources	Totals (PT1+PT2)	Languages estimations (%)
Dynamic resources collected from feeds (text-based feeds and feed items)	WWW (UvA)	> 25,000	Not specified
Constrained Focused Crawl (Web pages)	WWW (UvA)	>130,000	ITA (~3,5%), SPA (~10,6%), GER (~3,8%), ENG (23%), POL(~11,3%), NDL(~16%) Wikipedia (all languages) (~32%)
Metadata - Web	BVCM	>2,600	SPA (100%)
Metadata - Text	BVCM National Library Austrian National Library OAI (Alinari), MICHAEL, Alinari, BandG	>280,000	SPA (<0,05%), ENG(>70%), GER (<0,2) ITA(<0,03%) POL(<13%) NED(<0,09%)
Still images	Alinari, AISA (Alinari)	> 12,000	ITA (~83%), SPA (~41%), GER (~41%), ENG (100%), POL(~41%), NDL(~41%)
Video	BandG, Alinari, TecheRAI (Alinari)	>570	NDL (>90%) ENG (<10%)
Audio podcasts	WWW (UvA)	>100 h	ITA (~14%), SPA (~8,7%), GER (~31%), ENG (22%), NDL(~24%)

Fig. 2. Cumulative contents collected (PT1+PT2) and language distribution. Brackets, in the language column, indicate the percentage of the content available in the specific language with respect to the total data volume available

performance and usability. In particular, we conducted a laboratory-based and a user-centred evaluation based on field trials. The evaluation of the different components developed in the project showed that the technical solutions adopted provided an advance w.r.t. existing systems. Furthermore, in order to get some basic idea of how fast, and how many concurrent users the search service can support, some simple load testing, was performed. The search service and the content cache proxy component were deployed in a Tomcat instance on a desktop test machine (an Intel Core Duo @ 2.66GHz with 2Gb of RAM). The tests, conducted for the most expensive search tasks (image and video search), showed that the prototype is able to efficiently process up to 40 users even with the limited computer resources used, while for more than 100 concurrent searches approximately 10% of requests time out. The Field Trials (FT) were conducted in

three different application settings, simulating the day-by-day use of the system. The selected application areas were:

- Education (20 users), for the use of MultiMatch in the Humanities
- Tourism (9 users), finding cultural events related to locations
- Cultural Heritage (26 users), by evaluating the use of MultiMatch in three professional environments.

Each Field trial (FT) attendee performed a set of tasks, (i.e. sequence of operations) chosen according to the User Group to which the attendee belonged.

Finally, as part of the evaluation, users of WIND Libero portal (with 28 million registered users) have been invited to use the MultiMatch Search engine. Users had to fill a questionnaire and were interviewed. They provided comments regarding both performance problems and also comments regarding the new functionality offered and the potential of the system. Most users felt that MultiMatch compares very well with other tools they are currently using and in many cases offers solutions that are not actually available. Users found the interface intuitive and, although participants had some problems with the use of the language tools, they did see a lot of potential for them and appreciated the functionalities offered. Improving the way multilingual search is supported by the system will be a priority in future research.

5 Conclusion

Multimatch manages a lot of various file types with highly different internal structure and complex semantics, it also generates various types of relations between the resources and it deals with several languages and user needs. The project was completed on October 31st, 2008 and all planned objectives were achieved. The MultiMatch project involved 11 partners: Istituto di Scienza e Tecnologie dell'Informazione - Consiglio Nazionale delle Ricerche (ISTI-CNR), University of Sheffield, Dublin City University, University of Amsterdam, University of Geneva, Universidad Nacional de Educacion a Distancia, Fratelli Alinari Istituto Edizioni Artistiche SpA, Netherland Institute for Sound and Vision, Biblioteca Virtual Miguel de Cervantes, OCLC PICA, WIND Telecomunicazioni SpA. The project website is active (http://www.multimatch.org) and an online demo of the MultiMatch prototype can be accessed by registered users (free registration available).

Acknowledgement
Work partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD - project MultiMATCH contract IST- 033104. The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

1. Heritrix: Heritrix.internet archive's web crawler project (2005) http://crawler.archive.org/.
2. Nutch: Nutch. open source web-search software (2005) http://lucene.apache.org/nutch/.
3. OAI: Open archives initiative (2005) http://www.openarchives.org/.
4. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery (1999)
5. Chakrabarti, S.: Mining theWeb: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann (2002)
6. Bruno, E., Moenne-Loccoz, N., Maillet, S.M.: Interactive video retrieval based on multimodal dissimilarity representation. In: 1st Workshop on Machine Learning Techniques for Processing Multimedia Content. (2005)
7. Clough, P.: Extracting metadata for spatially-aware information retrieval on the internet. In: In GIR-05 Workshop at CIKM 2005. (2005)
8. J.-Y. Nie, M. M.Simard, P.S., Durand, R.: Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In: In 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1999)
9. Wactlar, H., Christel, M., Gong, Y., Hauptmann, A.: Lessons learned from the creation and development of a terabyte digital video library (1999)
10. Daniela, P., Stephen, L., Micheline, B., Mark, S.: Which user interaction for cross-language information retrieval? design issues and reflections. J. Am. Soc. Inf. Sci. Technol. **57**(5) (2006) 709–722

Integrating Multi-dimensional Information Spaces

Kostas Saidis and Alex Delis
{saiko,ad}@di.uoa.gr

Department of Informatics and Telecommunications
University of Athens, 157 84, Athens, Greece

1 Introduction

Contemporary digital libraries (DLs) and related systems, including but not limited to institutional repositories, e-publishing platforms and cultural heritage systems, face the need to manage diversely structured information spaces. The latter are comprised of multi-faceted digitized and/or born-digital content, such as intellectual works, scholarly information, institutional or personal archives, cultural heritage material and even user-generated content such as blogs and video casts. Although such information spaces may –and in practice, actually do– originate from heterogeneous sources, including XML repositories or custom database solutions, the increased connectivity offered by the expansion of the Internet, raises the challenge to inter-connect such heterogeneous sources and integrate their information spaces in order to enable users to share, reuse, refine and extend them in varying application contexts.

In this paper, we view VLDLs as DLs that manage "large information spaces", not only in terms of the volume of content they can manage but also in terms of the diversity of material and the heterogeneity of sources they can support. Under this perspective, the information integration/interoperation requirements mentioned above are of significant importance to VLDL development. As we discuss herein, *VLDLs are dominated by the multi-dimensional nature of information space management*, as comprised of information discovery, access, conceptualization and utilization options. In particular, the ability to extend information spaces' management options in the aforementioned dimensions is strongly connected to the ability to integrate these spaces. Based on this, we identify several essential design issues of a common information space management infrastructure. This infrastructure will add value to VLDLs by simplifying –and automating as much as possible– complex information space integration tasks.

2 Rationale & Motivating Example

Making systems interoperate and integrate their information spaces is hard, because, in practical terms, different systems develop different views of digital content for different purposes, as Figure 1 shows. Viewing information space integration as a process roughly comprised of the following steps:

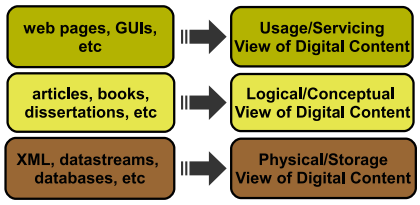


Fig. 1. Multiple views of digital content

a. information discovery: systems discover or "learn about" the existence of each other.

b. information identification/access: systems can unambiguously identify individual elements of their information spaces, while they are also provided with a means to access such elements.

c. information utilization: systems can synthesize the elements of their information spaces to accommodate to their particular service provision requirements.

it is rather clear that to perform the above steps requires dealing with a plethora of information discovery, access, conceptualization and utilization options supported by involved systems. Indeed, due to the multi-dimensional nature of digital content management, each system supports its own *information access* options to render its own *information conceptualizations* required to answer its particular *information utilization and usage* needs. Thus, when integrating information spaces, we practically need to extend involved systems in terms of multiple information management dimensions to:

- support new information discovery mechanisms,
- access new kinds of information sources,
- augment the system with new content usage scenarios
- revisit existing information conceptualizations

Realizing such interdependent and crosscutting extensions can be as complex and costly as adding a brand new set of features to a system, often "breaking" its existing design. However, cost-effectiveness is crucial for information integration/interoperability, since the latter is about "*enabling information that originates in one context to be used in another in ways that are as highly automated as possible*" [6]. Therefore, the success of an approach to integrate information spaces depends highly on *the level of automation achieved*, as an approach which involves the manual execution of costly and complex tasks is most likely to be ineffective in practice.

The Web, for example, fully automates information identification and access through the use of URLs and HTTP respectively. Information discovery is automated by value-added search services such as Google, while the Web's information utilization options, although automated, are based on a rather limited "document-based" conceptualization that adheres to a publish/consume paradigm. Technologies such as Web Services [8] and the Semantic Web [5] aim at enhancing the limited discovery and utilization options offered by the tra-

ditional Web. On one hand, Web Services can be used to discover and utilize "live" content comprised of both data and code. On the other, the Semantic Web extends the "document-based" nature of the traditional Web with the addition of semantic annotations to Web resources. Such annotations can then be used to enhance Web resource utilization in various contexts.

Although the Web is the world's largest interoperable information space, similar, if not identical, information discovery/identification, access, conceptualization and utilization issues arise in integrating smaller-scale information spaces. For example, our Pergamos DL offers a uniform platform for documenting, preserving and publishing the digital material of University of Athens [4]. Pergamos has been in production use for more than 4 years, currently hosting about 300,000 items exceeding 1 TB of space. These items are grouped in various collections that originate from digitization projects of various University departments, while upon their completion, collections are published through Pergamos' web-based front-end [7]. However, Pergamos is not the only DL system available at the University, as many departments have deployed various assorted DLs over the years, comprised of diverse collections held in heterogeneous storage solutions. For instance, there is *Anthemion*, a client-server database application hosting digitized books of the early 1900's. There is also a Lotus Domino application hosting theses and dissertations of the Faculty of Law. In this context we face the following dual challenge. On one hand we need to extend Pergamos to support new digital collections, as new digitization projects emerge. On the other, we need to make Pergamos interoperate with *Anthemion* and the Domino-based "legacy" DLs.

Any successful system will, sooner or later, need to extend its multiple information space management options. For example, Pergamos has to access Domino-based dissertations and Anthemion's database-oriented books in order to publish them using its existing web-based front-end. To realize this requires supporting new kinds of (a) information sources, extending Pergamos' information access options, (b) content items, namely, books and dissertations, thus extending Pergamos' information conceptualization options.

Moreover, VLDLs should share but also reuse, refine and/or extend their information spaces. For instance, in Pergamos we need to integrate information spaces that goes beyond sharing read-only digital content representations. In particular, we need Pergamos back-end system, used by our libraries staff to document content items through web-based forms, to be able to access, yet also update and modify the Domino and Anthemion collections. This yields an additional need to extend existing content utilization options, as Pergamos back-end should be made to update these newly added collections, not only access them in a read-only fashion. Finally, Pergamos' information discovery options should be also extended in order to realize cross-DL searching and retrieval.

This highlights the challenges that arise in contemporary DLs and related systems caused by the need to amalgamate their information spaces. This need has a drastic affect on DL systems, as it imposes extensions in multiple of their information space management options. Therefore, it is rather critical to supply

systems with an infrastructure for performing such extensions in a straightforward and cost-effective fashion.

3 Multi-dimensional Information Space Management

Figure 2 shows a VLDL managing a multi-dimensional information space. The former consists of the services, which, instrumented by the VLDL's application logic, synthesize the information space in terms of various information access, conceptualization, utilization and discovery options.

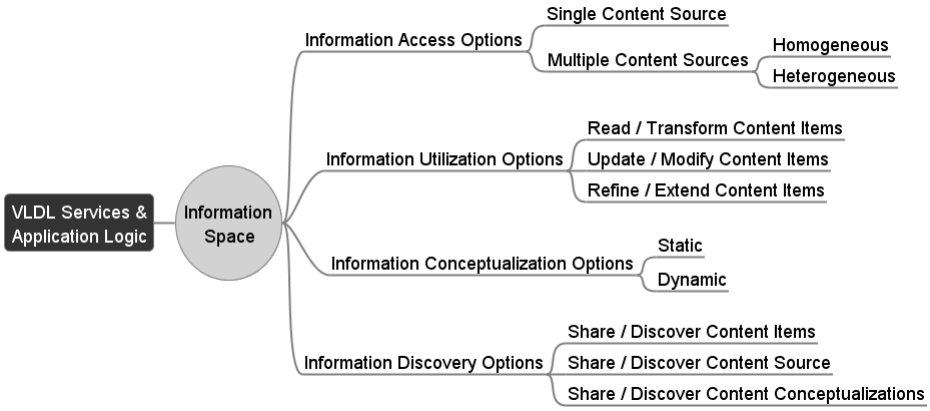


Fig. 2. Multi-dimensional Information Space Management Options

The figure can be viewed as a taxonomy of a DL's information space management features, identifying several essential characteristics of its overall functionality. For example, in a metadata harvesting scenario, a central DL employs a harvesting protocol to gather scholarly information from various sources. Although individual content sources may be heterogeneous, the use of a common high-level harvesting protocol hides such details from the central DL, thus enabling the latter to treat all sources in a unified manner, as if they were homogeneous, advancing interoperability. Moreover, given that this scenario involves scholarly material, the central DL can issue a static information conceptualization of "article objects".

Clearly, the information space management options of Figure 2 are highly coupled to each other, reflecting the complexity of interoperable DLs. In a technical context, the addition of value offered by the use of common information space management infrastructure resides in enabling DL application developers and designers to treat the four information space management dimensions of Figure 2 independently of each other. This will offer standard and effective mechanisms to extend a DL's information discovery, access, utilization and conceptualization options, ideally without (a) affecting the functionality of the DL

system in the remainder dimensions, (b) requiring any complex and costly DL system code modifications and (c) causing a DL shutdown and/or restart. Placing the infrastructure between a DL's application logic and its information space, in the remainder of this section we identify some essential design issues that help achieve the aforementioned goal.

3.1 Information Access Options

We use the information access dimension to render a logical view of a system's content access capabilities, reflecting *the fundamental functionality to contact a content source and fetch its items*. The system will employ a specific "data access machinery" to contact a content source, a set of access mechanisms and related tools, including involved protocols, XML handling libraries or database access components. From a system design perspective, we view sources that are accessed using a common "data access machinery" as homogeneous and those which require different machineries as heterogeneous. We identify the following three major information access options, where a system may operate atop:

1. *a single content source*: all the operations related to information storage and retrieval are performed via a single source. For example, a personal DL is expected to employ a local content store solution, while a web-based DL may use a more sophisticated multi-tier architecture. Yet, in both cases, the DL system in question uses a single "data access machinery".
2. *multiple homogeneous sources*: the system employs several homogeneous sources to meet its application logic requirements. One such example is the metadata harvesting system described before. Another example may be a load balancing scenario, where a DL application server redirects access requests to a set of identical content sources based on the system's load. In both examples, the content sources are homogeneous, as the DL system in question uses a single "data access machinery" to contact all of them in a unified manner.
3. *multiple heterogeneous sources*: the DL system has to combine the usage of various "data access machineries" to contact a variety of content sources, in order to answer the needs of its application logic. An example of such a case is our Pergamos front-end, which has to manage content that originates from the Pergamos XML repository, the Domino-based theses and dissertations and Anthemion's custom database.

The infrastructure should enable DLs to operate atop multiple heterogeneous sources in a cost-effective manner. For example, should we consider a DL that supports a single content source, the challenge here is to progressively extend its access options to include more sources, as new content access needs emerge. To this end, the infrastructure should offer a unified, system-wide *Content Access API*, used to (a) register new sources in the information space, dealing with their specific "data access machineries" and (b) allow DL services to fetch underlying content items uniformly, regardless of their location and storage details.

3.2 Information Utilization Options

We use the information utilization dimension to refer to the essential operations a system performs on its content items, namely:

1. *read/transform content items*: this is a fundamental operation, which, as mentioned before, is offered by the unified *Content Access API*, allowing DL services to to fetch any content item originating from any content source. Services can then synthesize the data held in the item to meet any read-only service provision requirement, such as dissemination or web publishing.
2. *update/modify content items*: except of the ability to fetch content items, DL services may also need to update/modify content items. The infrastructure should answer such a need by offering an appropriate *Content Update API*, providing a unified, system-wide support for performing update/modify operations on the data held in the content items. For example, our Pergamos back-end cataloging UI should use this API to generate web forms for allowing catalogers to create, document, update and modify content items. The role of this API is to assist DL implementors to realize high-level information update services, such as cataloging web forms, administration UIs, ingestion or migration services, while maintaining the ability to operate atop multiple heterogeneous sources.
3. *refine/extend content items*: this is the most advanced case, as it refers to the ability to modify the structure of content items, not only the data they hold. Ideally, the ability to build upon existing content items to generate new ones should be also given. Content item refinement and extension is strongly coupled to the conceptualization options supported by the DL, as discussed in the next section.

3.3 Information Conceptualization Options

Consider a system such as *Anthemion* which manages "book" items. Although we may all share a common perception of the "book" concept, in a technical context, "book" items may have different conceptualizations in different systems. For example, in a context where books are digitized, it is not uncommon to represent such "books" as compound entities comprised of "pages". In another context, where a book's content is born-digital, as in the cases of e-books or other similar web material, it is rather common to represent "books" in a more simple fashion. Finally, in a scholarly information context, a conference's proceedings may be realized in terms of "book" items comprised of "articles".

Since DLs issue various conceptualizations for their content items, we use the information conceptualization dimension to render a high-level view of a system's conceptualization capabilities, where a system supports:

1. *static information conceptualizations*: *Anthemion*'s information space is comprised of "book" items, while the Domino application manages "dissertation" items. The conceptualizations supported by these DLs are *static*, in the sense that they are hard-coded in the DL application code. Should

we consider that a need to support some new types of content items occurs, we cannot make such systems answer this need without performing costly modifications to their source code. The same stands when a need to modify the structure of existing "book" and/or "dissertation" items occurs.

2. *dynamic information conceptualizations*: the infrastructure should follow a more *laissez faire* approach on expressing and handling information conceptualizations, as ideally, it should be able to support diverse content conceptualizations dynamically. Viewing content conceptualizations as definitions of the structure and relationships of content items, the infrastructure should offer the means to support varying kinds of such definitions, which can be added into the information space in a dynamic fashion. DL services could then use these definitions to identify the conceptualizations pertinent in the information space and adjust to their requirements dynamically. This way, the system will also adapt more effectively when the information space needs to be augmented with a new type of content items.
   Moreover, given that conceptualization definitions will be independent of any particular content storage and content utilization details, the infrastructure could also employ an inheritance mechanism for building upon existing definitions to generate new ones, allowing users and services to refine and extend conceptualizations in various information spaces.

3.4 Information Discovery Options

In terms of information discovery, it is a commodity for a system to employ a search utility to assist users –or other systems– to search its information space and retrieve items that match specific criteria. Search functionality can be primarily viewed as a helper for simplifying information space usage, yet, it is also coupled to the information access options supported by a system, as indexing facilities usually accompany data storage systems. For example, in a scenario where various content sources participate in a DL federation, one may use a centralized indexing approach where all content is indexed in a central indexing facility. Another approach could distribute search queries to the individual indexing facilities that may be present in the network.

In the scope of this paper we place our focus on identifying the information that should be possible to share/discover using the infrastructure, namely:

1. *share/discover content items*: this refers to the main information discovery requirement, the ability to search for specific content items.
2. *share/discover content sources*: the infrastructure should also provide the ability to share and discover content sources. This will allow remote DLs to gain access to their underlying content sources, further fostering their integration. For example, a DL may publish an index of its content sources along with a wrapper of its *Content Access API* –e.g., a web service. This will allow remote systems to reuse the DL's information access options and contact the DL's sources to access hosted items.
3. *share/discover content conceptualizations*: the ability to share/discover definitions of content conceptualizations should be provided, as it will enable

DLs to discover new kinds of content items. For example, in a fashion similar to the above content source sharing scenario, a DL may publish its content conceptualization definitions, offering this way access to the structure and relationships of its content items. This will enable remote DLs to use these definitions in various usage scenarios –e.g., to build upon existing definitions to generate new conceptualizations, or to use the definitions to generate a content item browsing service over the whole DL network.

Thus, the infrastructure should need to provide three information discovery facilities: one for searching content items, one for discovering content sources and one for sharing content conceptualizations.

4 Closing Remark

In this paper, we have identified several essential requirements for supplying DLs with a common information space management infrastructure. Firstly, the infrastructure should treat information spaces as multi-dimensional information management contexts, comprised of information discovery, access, conceptualization and utilization options. Secondly, in order to simplify integration and interoperation of information spaces, it should automate the extension of these options independently of each other. We believe that such an approach can assist the goal of the DL research community, as expressed in efforts such as the DELOS reference model [1], the 5S model [3] and the OAIS reference model [2], to offer a unified DL foundation.

References

1. L. Candela, D. Castelli, P. Pagano, C. Thanos, Y. Ioannidis, G. Koutrika, S. Ross, H.-J. Schek, and H. Schuldt. Setting the Foundations of Digital Libraries: The DELOS Manifesto. *D-Lib Magazine*, 13(3/4), March/April 2007. [doi:10.1045/march2007-castelli].
2. Consultative Committee for Space Data Systems (CCSDS). Reference Model for an Open Archival Information System (OAIS). Blue Book, Issue 1, `http://public.ccsds.org/publications/archive/650x0b1.pdf`.
3. M. Gonçalves, E. Fox, L. Watson, and N. Kipp. Streams, Structures, Spaces, Scenarios, Societies (5s): A Formal Model for Digital Libraries. *ACM Transactions on Information Systems (TOIS)*, 22(2):270–312, 2004.
4. K. Saidis, G. Pyrounakis, M. Nikolaidou, and A. Delis. Digital object prototypes: An effective realization of digital object types. In *ECDL '06: Proceedings of the 10^{th} European Conference on Digital Libraries*, Alicante, Spain, September 2006.
5. N. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.
6. The International DOI Foundation. The DOI Handbook. Edition 4.4.1, October 2006, [doi:10.1000/182].
7. University of Athens. Pergamos Digital Library. `http://pergamos.lib.uoa.gr/`.
8. *Web Services Activity*. World Wide Web Consortium. Available at `http://www.www.org/2002/ws/`.

Improving Similarity Search in Face-Images Data

Pedro Chambel[1], Fernanda Barbosa[1]

[1] FCT-Universidade Nova de Lisboa, Portugal
{pms17299@fct.unl.pt, fb}@di.fct.unl.pt

Abstract. Similarity search involves finding all the face-images in a database, which are similar to a desired face-image, based on some distance measure. Comparing the desired face-image to all the face-images in a large dataset is prohibitively slow. If face-images can be placed in a metric space, search can be sped up by using a metric data structure. In this work, we evaluate the performance of range queries with metric data structures (LAESA, VPtree, DSAT, HDSAT1, HDSAT2, LC, RLC and GNAT) when the metric spaces are face-images data with the Euclidean distance. The experimental results show that all data structures reduce the ratio between the number of distances computed and the database size. Moreover, the LAESA has the best performance in the majority of the experimental cases, but the RLC competes with the other metric data structures, and has the best results when compared with the other dynamic metric data structures.

Keywords: similarity search; metric data structures; image processing

1 Introduction

With the rapid increase in the use of digital technology, large amounts of image data sets will soon be accumulated. Face-images browsing is based on the concept of similarity between face-images, i.e. searching face-images which are very similar or close to a given face. Each face-image is represented as a vector of numeric properties (features) extracted from the contend-based image. The similarity between two images is associated with a function, which measures the distance between their feature vectors. When this function is metric, the set of face-images defines a metric space.

Most of the actual work in face-images is about face detection and face recognition. In these works, the similar searching for a given face-image leads to an exhaustive search in the dataset, so the response time will be very long and the search will became ineffective. For this reason, it is necessary to introduce new techniques that can deal with this problem efficiently.

In order to have efficient similar searching in metric spaces, several metric data structures have been proposed [1,2]. These data structures partition the database based on the distances between a set of selected objects and the remaining objects. Space partitions seek to minimize the exhaustive search, i.e. at search time, some subsets are discarded and others are exhaustively searched. The distance-based indexing method

may be pivot-based or cluster-based [1]. Some of the data structures using the pivot-based method are the VP-Tree [3] and the MVP-Tree [4]. There are variants of the pivot-based method, used in LAESA [5]. Some of the data structures using cluster-based method are the GNAT [6], the HDSAT [7], the LC [8] and the RLC [9]. The RLC was already evaluated in different application domains [10, 11, 12].

Our main goal is to evaluate the use and the efficiency of similar searching with metric data structures in face-images databases with Euclidean Distance. This study involves four databases of face-images: Faces94 [13], Jaffe [14], Yalefaces [15] and AT&T [16], and comprises 8 metric data structures: LAESA, VPtree, DSAT, HDSAT1, HDSAT2, LC, RLC and GNAT.

The rest of the paper is structured as follows. In Section II, we recall some basic notions on range query in metric spaces. Section III is devoted to the characterization of the metric space over face-images data. Then Section IV reports on the experimental results. Conclusions and future work are drawn in Section V.

2 Range Queries in Metric Spaces

A metric space is a pair (U,d), where U is a set of objects, called the universe, and d: U x U $\Rightarrow \Re^+_o$ is a function, called distance, that satisfies the three following properties: (1) strict positiveness: $d(x,y) \geq 0$ and $d(x,y) = 0 \Leftrightarrow x = y$; (2) symmetry : $d(x,y) = d(y,x)$ and (3) triangle inequality: $d(x,y) \leq d(x,z) + d(z,y)$.

A database over a metric space (U,d) is a finite set $B \subseteq U$. The problem raised by range queries is to yield the set of all database objects whose distance to a given object does not exceed a certain amount. Formally, given a database B over a metric space (U,d), a query point $q \in U$, and a query radius $r \in \Re^+$, the answer to the range query (q,r) is the set $\{x \in B \mid d(x,q) \leq r\}$.

Metric data structures seek to minimize the number of distance computations performed in range queries. During the computation of a range query (q,r) in a database over a metric space (U,d), triangle inequality and symmetry are used to discard elements of the database without computing the associated distance to the query object. Given a query element q and a query radius r, an element x may be left out without the need for evaluating d(q,x). This will arise if there is an object o where $|d(q,o) - d(x,o)| > r$. In this case, it is not necessary to compute d(q,x) since we know that $d(q,x) > r$, based on the triangle inequality.

It is important to remark that the range queries are hard to compute in high dimension metric space [2]. It is well known that the metric space dimension grows with the mean and decreases with the variance.

3 Face-Images Metric Space

Our experiments involve four face-images databases, which were already used in related works [17, 18, 19]. The databases used are:

- Faces94 – This database is available in [13]. The images are stored in 24 bit RGB, JPEG format and the size of each image is 180 x 200 pixels.

- Jaffe – This database is available in [14]. The images are stored in TIFF format and the size of each image is 256 x 256 pixels.
- Yalefaces- This database is available in [15]. The images are stored in GIF format and the size of each image is 320 x 243 pixels.
- AT&T- This database is available in [16]. The images are stored in PGM and the size of each image is 92x112 pixels, with 256 grey levels per pixel.

In these databases, all the images are frontal face-images of different individuals in different facial expressions. In table 1 we present the size of each database.

3.1 Face-Image Representation

Each face-image is represented by a feature vector, which describes the face according to a training set of face-images. The method used to extract this feature vector was based on the method of Eigenfaces. The steps involved in creating a set of eigenfaces are:

- Define a training set: each image is seen as one vector, simply by concatenating the rows of pixels in the image. So an image with r rows and c columns is therefore represented as a vector with r x c elements. All images in the training set are stored in a single matrix T, where each row is an image;
- Calculate the average image and subtract it from each image in T;
- Calculate the eigenvectors and eigenvalues of the covariance matrix S. The eigenvectors of this covariance matrix are called eigenfaces;
- Choose the principal components by keeping the eigenvectors with the largest associated eigenvalue.

The fundamental idea of this method is to project the face-images on the eigenfaces created. So, each face-image is a vector of features $S = <f_1, \ldots, f_n>$, where n is less than or equal to the size of the training database. In our experiment, the size of the training set is 25 face-images. For further details of the method of Eigenfaces, the reader is referred to [20].

3.2 Similarity between Face-Images

In our experiment, the similarity between two face-images is based on the similarity between the associated feature vectors, which is computed with the Euclidean distance. Let S = <f1s, …, fns> and T = <f1t, …, fnt)> be feature vectors associated with two face-images S and T. The Euclidean distance between S and T, denoted by ED(S,T), is defined by:

$$ED(S,T) = \sqrt{\sum_{i=1..n} (f_{is} - f_{it})^2}. \quad (\mathbf{1})$$

In order to study the metric spaces, we have computed the histogram of distances between any two face-images of each database. In Table 1, we present the mean and the variance of the histogram of distances for each database.

An immediate conclusion is that our metric spaces have different dimensions. The dimension is highest for the Jaffe database, where the quotient between the mean and the variance is 2.7710. The other metric spaces have lowest dimension.

Table 1. Database Size and Mean and Variance of the Histogram of Distances

Database	Size(images)	Mean	Variance	Mean/Variance
Faces94	3040	9137.49	7936650.53	0.0012
JAFFE	213	10749.96	15050638.74	2.7710
Yalefaces	165	19293.42	37809562.91	0.0005
AT&T	400	3008.78	617807.58	0.0049

3.3 Evaluation of Euclidean Distance in the Databases

The Euclidean distance was evaluated in the 4 databases with the ROC (Receiver Operating Characteristics) analysis, based on the true positive rate and the false positive rate. In all the databases, the results show that the false positive rate (irrelevant results) is less than 0.1 and the true positive rate (relevant results) is bigger than 0.5. The Euclidean distance shows better results with the Faces94 database and worse results with the Yalefaces database. For lack of space, this evaluation can not be presented here.

4 Evaluation of the Metric Data Structures

The goal of this section is to evaluate the behavior of range queries with metric data structures (LAESA, VPTree, DSAT, HDSAT1, HDSAT2, LC, RLC and GNAT) over face-images data with the Euclidean distance.

For each database, four files were generated. The smallest was used as the query set of faces and is composed by random faces from the database. In order to maintain the same number of different individuals in this set, we choose randomly 25% of the face-images associated to each individual. Table 2 presents the size of the query set for each database. The other three files are random permutations of the database. The justification for making use of three equal sets lies on the fact that the final shape of some data structures depends on the order in which the objects occur in the input of the construction algorithm.

In order to compute the range queries with the metric data structures we choose two different query radii for each database. The first one is 25% and the second is 50% of the database distances mean (see Table 2).

Table 2. Size of the Query Set and Query Radii for each Database

Database	Size(query set)	1st Query Radius	2nd Query Radius
Faces94	760	2284	4569
JAFFE	50	2687	5374
AT&T	80	752	1504
Yalefaces	30	4823	9646

For each database, we submitted the set of query faces associated with the database with the two radii. In Table 3, we present the average number of face-images retrieved in range queries for each query, and the associated percentage of the database size.

Table 3. Average number of face-images retrieved in range queries, and the associated percentage of the database size

Database	1st Query Radius		2nd Query Radius	
	Num	Percent	Num	Percent
Faces94	15.94	0.52%	109.79	3.61%
JAFFE	6.76	3.17%	21.94	10.3%
AT&T	2.35	0.59%	12.83	3.21%
Yalefaces	3.7	2.24%	11.43	6.93%

In each experimental case (a database, a query set and a query radius), we compute the average number of distance computations done for each face-image query. So, the results presented are the mean of the results obtained to query the three sets associated to the database.

4.1 Metric Data Structures Parameterizations

The eight metric data structures were parameterized in order to obtain the best results for each database:

- LAESA: The Linear Approximating and Eliminating Search Algorithm was parameterized with 44 prototypes for Faces94, 8 prototypes for JAFFE, 22 prototypes for AT&T and 23 prototypes for Yalefaces.
- VP-tree: This data structure does not have parameters.
- DSAT: The Dynamic Spatial Approximation Tree was parameterized with arity 3 for Faces94, Jaffe and Yalefaces, and with arity 5 for AT&T.
- HDSAT1: The Hybrid Dynamic Approximation Tree 1 was parameterized with arity 5 for Faces94, 6 for JAFFE and AT&T, and 9 for Yalefaces.
- HDSAT2: The Hybrid Dynamic Approximation Tree 2 was parameterized with arity 8 for Faces94, 6 for JAFFE and AT&T and 9 for the Yalefaces.
- LC: The List of Clusters we was parameterized with cluster radius 2995 for Faces94, 7130 for JAFFE, 1400 for AT&T and 8795 for Yalefaces.
- RLC: The Recursive List of Clusters was parameterized with cluster radius 3575 and array capacity 20 for Faces94, cluster radius 6750 and array capacity 11 for JAFFE, cluster radius 1840 and array capacity 9 for AT&T and cluster radius 9450 and array capacity 11 for Yalefaces.

4.2 Experimental Results

Figures 1 and 2 show, for each database, the average number of distance computations done with each query with the two query radii.

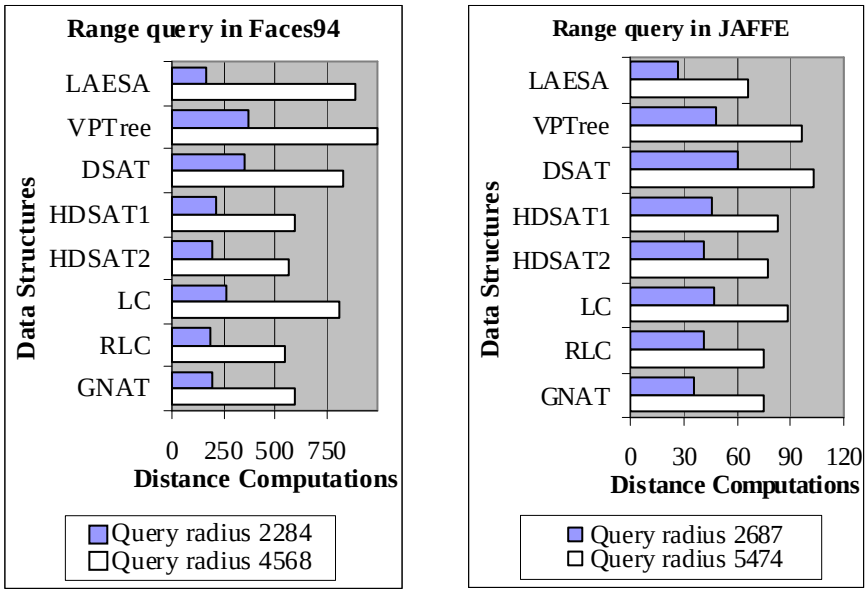


Fig. 1. Average number of distance computations in Faces94 and JAFFE databases

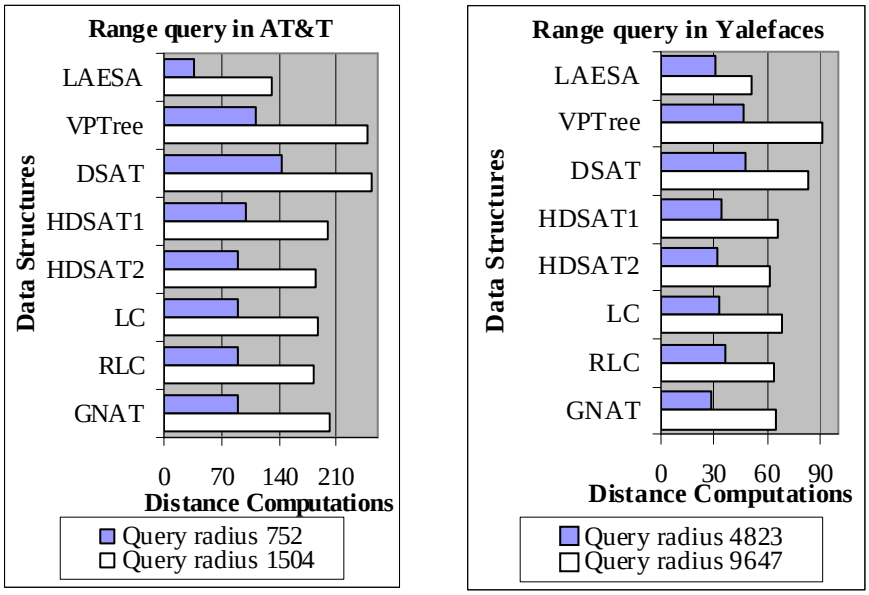


Fig. 2. Average number of distance computations in AT&T and Yalefaces databases

These graphics show that LAESA outperforms all the data structures, except in the Faces94 database with query radius 4569, where RLC achieves the best result, and in Yalefaces database with radius 4823, where the GNAT is the best.

When we compare the four dynamic metric data structures (DSAT, HDSAT1, HDSAT2 and RLC) we conclude that, in the majority of the experimental cases (5 of 8 cases), RLC is the data structure with the best results.

The results from this experiment show that all the metric data structures reduce the ratio between the number of distances computed and the size of the database. Table 5 presents the percentage for each data structure and each database.

Table 5. The Percentage of Distance Computations According to the Database Size

Database	Faces94		JAFFE		AT&T		Yalefaces	
Query radius	2284	4568	2687	5474	752	1504	4823	9646
LAESA	5.38%	29.25%	12.39%	31%	8.86%	32.76%	18.52%	30,77%
VPTree	12.14%	32.71%	22.58%	45.4%	27.68%	62.01%	28.61%	54,67%
DSAT	11.72%	27.36%	28.67%	48.6%	35.96%	63.01%	29,04%	50,06%
HDSAT1	7.25%	19.5%	21.33%	38.91%	24.86%	49.6%	20.78%	39,7%
HDSAT2	6.57%	18.6%	19.6%	36.29%	22.63%	46.26%	19.25%	36,75%
LC	8.57%	26.76%	22.3%	41.71%	22.56%	46,7%	19,98%	41,23%
RLC	6.27%	18.17%	19.55%	35.48%	22.67%	45,83%	22,27%	38.76%
GNAT	6.5%	19.46%	16.64%	35.38%	22.44%	50.72%	17,44%	39.03%

5 Conclusion and Future Work

The need to speed the similar searching of face-images leads us to evaluate the performance of range queries with several metric data structures over face-images data. It is important to remark that there are few works which compare different techniques in the face-image domain.

With respect to the face-images data representation and the Euclidean distance (our metric space), we have good similar search results in these databases. However, we need to compare these results with the results obtained in face-images databases, where the data are described by local features (i.e., noise, eyes) and/or with others metric functions.

With respect to the efficiency of the range queries with metric data structures, the results leave us to conclude that the metric data structures speed the range query in the four databases. This conclusion is based on the observation that a lot of face-images are discarded without computing the associated distance to the face4-image query. The LAESA data structure has the best performance in all the databases, except in two experimental cases. In the majority of the experimental cases, the RLC data structure competes with the other metric data structures, and it is the best dynamic metric data structure in the majority of the experimental cases.

References

1. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, San Francisco (2006)
2. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J.: Searching in metric spaces. ACM Computing Surveys, 33(3), 273—321 (2001)
3. Yianilos, P.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: 4th Annual SIAM Symposium on Discrete Algorithms, pp. 311—321, ACM Press, USA (1993)
4. Bozkaya, T., Ozsoyoglu, M.: Distance-based indexing for high-dimensional metric spaces. In: SIGMOD International Conference on Management of Data (SIGMOD'97) , pp. 357—368, ACM Press, New York (1997)
5. Micó, M., Oncina, J., Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm. Pattern Recognition Letters, 15(1), 9—17 (1994)
6. Brin, S.: Near neighbor search in large metric spaces. In: 21st International Conference on Very Large Data Bases (VLDB'95), pp. 574—3584, Morgan Kaufmann, Zurich, Switzerland (1995)
7. Arroyuelo, D., Muñoz, F.; Navarro, G., Reyes, N.: Memory-adaptative dynamic spatial approximation trees. In: the 10th International Symposium on String Processing and Information Retrieval (SPIRE) (2003)
8. Chávez, E., and Navarro, G.: A compact space decomposition for effective metric indexing. Pattern Recognition Letters, 26(9), 1363—1376 (2005)
9. Mamede, M.: Recursive Lists of Clusters: a dynamic data structure for range queries in metric spaces. In 20th International Symposium on Computer and Information Sciences (ISCIS 2005) , pp. 843—853, Springer-Verlag, Berlin, Germany (2005)
10. Mamede, M., Barbosa, F.: Range Queries in Natural Language Dictionaries with Recursive Lists of Clusters. In: 22th International Symposium on Computer and Information Sciences, IEEE Xplore, doi:10.1109/ISCIS.2007.4456857 (2007)
11. Barbosa, F.: Similarity-based retrieval in high dimensional data with Recursive Lists of Clusters: a study case with Natural Language Dictionaries. In: International Conference on Information management and engineering (ICIME 2009), IEEE Computer Society, ISBN: 978-1-4244-3774-0 (2009)
12. Barbosa, F., Rodrigues, A.: Range Queries over Trajectory Data with Recursive List of Clusters: a case study with Hurricanes data. In: *Geographic Information Science Research UK (GISRUK 2009)*, UK (2009)
13. Faces94 database : http://cswww.essex.ac.uk/mv/allfaces/
14. Jaffe database: http://www.kasrl.org/jaffe.html
15. Yalefaces database: http://cvc.yale.edu/projects/yalefaces/yalefaces.html
16. AT&T database: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
17. Allam, H., Rahman, F., Tarnikova, Y., Hartono, R.: A Pair-wise Decision Fusion Framework: Recognition of Human Faces. In: International Conference on Information Fusion (ISIF 2003),pp. 1484-1489, IEEE Xplore, ISBN: 0-9721844-4-9 (2003)
18. Lyons, M., Akamatsu S.: Coding Facial Expressions with Gabor Wavelets. In: 3rd IEEE International Conference on Automatic Face and Gesture Recognition (AFGR 1998), pp. 200—205, IEEE Xplore, ISBN: 0-8186-8344-9 (1998)
19. Aly M.,: Face Recognition using SIFT Features. Available on http://www.vision.caltech.edu/malaa/research/aly06face.pdf
20. Turk M., Pentland A.: Face Recognition using Eigenfaces. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586—591 (1991)

Improving Query Results with Automatic Duplicate Detection

Irina Astrova

Institute of Cybernetics, Tallinn University of Technology, Akadeemia tee 21,
12618 Tallinn, Estonia
irinaastrova@yahoo.com

Abstract. Ontology-based data integration poses significant challenges. One is that an ontology used as a global reference model during the ontology-based data integration can contain duplicated attributes, which can easily lead to improper query results. This problem arises when merging similar or overlapping information from ontologies extracted from distributed digital libraries into a single global ontology. To solve the problem, we propose a novel context-based approach that analyzes a workload of queries over the single global ontology to automatically calculate (semantic) distances between attributes, which are then used for duplicate detection. We present experimental results to demonstrate the quality of our approach.

Keywords: Ontology-based data integration, duplicate detection, market basket analysis, ICD algorithm.

1 Introduction

Semantic heterogeneity is the ambiguous interpretation of terms describing the meaning of data in heterogeneous resources such as distributed digital libraries [1]. This is a well-known problem in data integration. A recent solution to this problem is to use ontologies; this is called *ontology-based data integration* [2, 3, 4].

There are three main advantages in ontology-based data integration:

1. Ontologies provide a rich-predefined vocabulary that serves as a uniform interface for querying over distributed digital libraries, the interface that is independent of the underlying database schemata.
2. The knowledge represented by ontologies is sufficiently comprehensive to support translation of all relevant data.
3. Ontologies support consistent management and recognition of inconsistent data [3, 5].

The main problem with ontology-based data integration is that ontologies are heterogeneous themselves [6]. In particular, they can be expressed in different ontology languages, each with its own syntax, semantics and expressivity. Even using the same ontology language does not solve the semantic heterogeneity problem, because ontologies can contain duplicated attributes.

As an example, consider a user who submits the following query against the Wikipedia infobox ontology [7]: "Which performers were born in Chicago?" In response to this query, the query-answering system will return only one result (viz. Michael Ian Black). However, if it were known that *actor* and *comedian* are subclasses of *performer* and that their attributes *birthplace*, *birth place*, *city of birth*, *place of birth* and *origin* are duplicates of *performer*'s *location*, the query-answering system could return 163 additional results. Thus, the recall of query results can be greatly improved by detecting duplicated attributes.

Duplicate detection may be manual, automatic, or both. Traditionally, duplicate detection is performed by humans (e.g. domain experts and users): "Humans do it better" [8]. Many ontology languages provide the means to specify duplicates. E.g. OWL [9] has a construct *sameAs*. However, humans cannot cope with a large set of attributes. As a result, manual duplicate detection tends to be slow, tedious and inefficient, and does not work on a large scale. Therefore, there is a need for automatic duplicate detection.

2 Related Work

Most research focuses on identifying similar attributes, with some research devoted to detecting duplicates. However, the approaches to identifying similar attributes are generally the same as those used to detect duplicates (i.e. very similar attributes). Over the past two decades, researchers in both academy and industry have proposed various approaches to identifying similar attributes. These approaches can be categorized as:

1. **Term-based (or linguistic) approaches** where *two attributes are considered to be similar if their names (i.e. terms) are similar* [8, 10].
2. **Value-based (or extensional) approaches** where *two attributes are considered to be similar if their values are similar* [8, 10].
3. **Structure-based (or taxonomic) approaches** where *two attributes are considered to be similar if their structures (i.e. taxonomies) are similar* [8, 10].
4. **Context-based approaches** where *two attributes are considered to be similar if their contexts are similar* [10].
5. **Hybrid approaches** that combine two or more approaches from the first four categories to minimize false positives (i.e. dissimilar attributes that appear similar) and false negatives (i.e. similar attributes that appear dissimilar) [8, 10].

The approaches can also be categorized based on the information they use for identifying similar attributes: domain knowledge from domain experts, the ontology itself (e.g. attribute names, values and taxonomies) and the past user interaction with the ontology (e.g. a workload of queries against the ontology and an edit history). E.g. Wu and Weld [7] proposed to use a history of changes made to the ontology and analyze this information to detect duplicates. There can be attributes in a class that are frequently renamed, or their values can be copied to one and the same attribute in another class. Such an edit history points to evidence that these attributes are

duplicates. However, the edit history must be recorded for a long time to minimize false positives and false negatives.

Furthermore, the approaches can also be categorized based on the techniques they use for identifying similar attributes: information retrieval, machine learning and graph theory techniques. E.g. Doan et al. [11], and Berlin and Motro [12] proposed to use machine learning techniques. These techniques require training data as input. In particular, users specify one or more positive examples, and the machine learning techniques attempt to retrieve similar attributes (based on the similarity of their values) through an iterative process of relevance feedback from the users. By contrast, Madhavan [13] proposed to use graph theory techniques for identifying similar attributes, which are organized in the taxonomy. These techniques assume that attributes appear to be similar if both those that precede them and those that succeed them do so.

3 Our Approach

A term-based approach can incur problems in situations where the same terms are used to name dissimilar attributes (i.e. homonyms) or where different terms are used to name similar attributes (i.e. synonyms). A value-based approach can incur problems in situations where similar attributes have no or few common values or where dissimilar attributes have many common values. A structure-based approach can incur problems in situations where similar attributes are not organized in the taxonomy or where the taxonomy is shallow. Since terms, values and structures are not sufficient criteria for identifying similar attributes, we propose to use a context-based approach where two attributes are considered to be similar if their contexts are similar. The main problem with this approach is how to identify similar contexts. We address this problem by adopting a similarity measure from market basket analysis.

Because domain experts are often unavailable and because the use of ontology itself for duplicate detection can be inefficient when the ontology contains many attributes, we propose to use the past user interaction with the ontology such as a workload of queries asked by users against the ontology. The workload typically contains a small set of attributes. Furthermore, the use of the workload does not require training data as input. This is a big advantage in the case where users submit queries against distributed digital libraries as training data are missing [14]. Moreover, gathering queries is much easier than gathering training data as profiling tools automatically record the queries into user profiles.

3.1 Market Basket Analysis

Market baskets are the sets of products bought together by customers in transactions. These may be the results of customer visits to the supermarket or customer online purchases in a virtual store. Typically, market baskets are represented as a binary matrix where rows correspond to transactions and columns to products. A row has a value of 1 for a column if the customer has bought the product in the transaction; otherwise, it is 0. The number of products and their price are ignored.

One of the most popular tasks of market basket analysis is to derive customer buying patterns. These patterns can be used to identify similar products. As an example, consider Coke and Pepsi. These two products appear dissimilar because they have few customers in common. However, it was observed that the customers of Coke and Pepsi bought many other products in common such as hamburgers, cheeseburgers, pizzas and chips. Based on this observation, Das and Mannila [15] defined the following similarity measure for products: *two products are considered to be similar if the buying patterns of their customers are similar*.

We adapt this similarity measure to attributes: *two attributes are considered to be similar if the querying patterns of their users are similar*. E.g. if it were known that there are many users who have asked about the birth place of actor together with the actor's name and birth date, and that there are many users who have asked about the origin of actor, again, together with the actor's name and birth date, we could conclude that attributes *birth place* and *origin* in a class *actor* are similar to each other.

We take advantage of relationship between querying patterns and user behaviors. In particular, users who have similar questions in mind submit similar queries, the queries with similar querying patterns [16]. E.g. when searching for a biography of actor, many users tend to ask about actor's name and birth date. Therefore, we propose to use a workload of queries asked by users against the ontology and analyze this information to derive querying patterns using data mining and pattern recognition techniques.

3.2 Assumptions

We assume that users do not ask about all attributes in the ontology at once. (This is by analogy with market basket analysis, which assumes that a market basket contains a small set of products from hundreds or thousands of products available in the supermarket or virtual store.) In the example above, the users have not asked about *actor*'s *nationality* and *marital status*. These are called *missing attributes*.

In addition, we assume that users understand the ontology well enough to submit queries that reveal the similarity between attributes, or the users intuitively know if the attributes are similar. E.g. there can be several recent queries in the workload by a certain user who may repeatedly ask about *actor*'s *birthplace*, *birth place*, *city of birth*, *place of birth* and *origin*.

3.3 Steps

Our approach goes through two basic steps:

1. Calculation of distances between attributes.
2. Detection of duplicates.

3.3.1 Calculation of Distances between Attributes

To calculate distances between attributes, we adopt the ICD (Iterated Contextual Distance) algorithm [15] from market basket analysis. The basic idea behind the ICD algorithm is to start with an arbitrary distance between attributes and use this distance to calculate a probability distribution of the attributes in the workload of queries, then use this distribution to recalculate the distance between the attributes. Since the calculation of a distance between attributes is circular, the ICD algorithm is iterative. A few iterations of the ICD algorithm (typically 5) produce a stable distance between attributes called an *iterated contextual distance*. This distance is between 0 and 1; 0 means that two attributes are completely similar and 1 means that they are completely dissimilar. Next, we present the ICD algorithm.

ICD ALGORITHM

INPUT: A workload of m queries over an ontology with n attributes.

OUTPUT: An n x n symmetric distance matrix in which an element standing in the i-th row and j-th column represents the iterated contextual distance between the attributes i and j.

1. **Construct a binary matrix.** Construct an m x n binary matrix *M* where rows correspond to the queries and columns to the attributes. Let *M*(i, j) be an element of the matrix *M* that stands in the i-th row and the j-th column. It has a value of 1 if the query i references the attribute j. Otherwise, it is 0.
2. **Construct a distance matrix.** Construct an n x n symmetric distance matrix *D* where both rows and columns correspond to the attributes. Let *D*(i, j) be an element of the matrix *D* that stands in the i-th row and the j-th column. It has a random value between 0 and 1 if $i \neq j$. Otherwise, it is 0.
3. **Construct query vectors.** Let R be a set of attributes in the ontology. For each attribute $A \in R$, let $r_A = \{t \mid M(t, A) = 1\}$ be a set of queries that reference the attribute A.
4. **Construct attribute vectors.** For each query $t \in r_A$, let $A_t = \{A \mid M(t, A) = 1\}$ be a set of attributes that the query t references.
5. **Construct probability distribution vectors**. For each attribute $A \in R$, let $V_A = \{f(t, A) \mid t \in r_A\}$ be its probability distribution vector, where f(t, A) is the probability distribution of the attribute A in the query t. It is calculated using formula (1):

$$f(t,A) = 1 - \prod_{A_i \in A_t} \left(1 - \frac{K(D(A_i,A))}{\sum_{C \in R} K(D(A_i,C))}\right) \quad (1)$$

   where K is a kernel smoothing function; e.g. K(X) = 1/(1+X).
6. **Calculate centroids of probability distribution vectors**. For each probability distribution vector V_A, let c_A be its centroid. It is calculated using formula (2):

$$c_A = \frac{1}{|V_A|} \sum_{f(t,A) \in V_A} f(t,A) \quad (2)$$

7. **Calculate distances between attributes.** For each pair of attributes $A \in R$ and $B \in R$ ($A \neq B$), let $D(A, B) = D(B, A) = |c_A - c_B|$, where c_A and c_B are centroids of V_A and V_B, respectively.
8. **Iterate:** Stop if the algorithm converges. Otherwise, go to Step 5.

3.3.2 Detection of Duplicates

To detect duplicates, we use a threshold; e.g. 0.20. Any two attributes with the iterated contextual distance less than this threshold are considered to be duplicates. For each pair of attributes $A \in R$ and $B \in R$ ($A \neq B$), let $S = \{(A, B) \mid D(A, B) < T\}$ be a set of duplicates, where $T \in [0, 1]$ is a threshold.

4 Experiments

We conducted experiments:

1. To evaluate the quality of the results produced by the ICD algorithm.
2. To find a relationship between this quality and the workload size.

4.1 Quality of Results Produced by the ICD Algorithm

In the experiments, we used a real ontology from the car sales domain [17]. This ontology did not require highly specialized domain knowledge, which simplified the task of finding people to participate in the experiments. Nor did the ontology contain many attributes, making it easy for the participants to query the ontology. In particular, the ontology had 17 attributes (both similar and dissimilar): *mileage, stock, stock number, color, miles, exterior, make, number of doors, body style, body type, drive wheels, horsepower, year, model, engine size, engine location,* and *price*.

We asked 2 people called "oracles" to rate each pair of attributes between 0 (completely similar) and 1 (completely dissimilar) using their intuition. These ratings constituted a gold standard.

In addition, we asked 5 people (different from the oracles) to provide us with queries that they would submit if they wanted to buy a car. Typically, the people asked about specific car characteristics in certain price range or explored various tradeoffs; e.g. price vs. horsepower. We collected a total of 37 queries, which constituted a workload. This workload was used as input to the ICD algorithm. When the ICD algorithm is run on the workload, it produced a binary matrix in Table 1 and a distance matrix in Table 2.

To evaluate the quality of the results produced by the ICD algorithm, we compared these results with the gold standard. The results appeared intuitive and reasonable. E.g. the ICD algorithm found the similarity between *miles* and *mileage* (a distance of 0.06), between *color* and *exterior* (a distance of 0.11), and between *stock* and *stock number* (a distance of 0.07).

Table 1. Binary matrix

	mileage	**stock**	**stock number**	**color**	**miles**	**exterior**	**…**	**price**
q1	1	0	0	1	1	0	…	1
q2	1	1	0	1	1	0	…	1
q3	1	0	1	0	0	1	…	1
q4	0	0	1	1	1	0	…	1
q5	0	1	0	0	0	1	…	1
q6	1	0	0	1	0	0	…	0
q7	0	0	0	1	1	0	…	1
q8	0	0	0	0	0	1	…	0
…	…	…	…	…	…	…	…	…
q37	0	1	0	0	0	1	…	1

Table 2. Distance matrix

	mileage	**stock**	**stock number**	**color**	**miles**	**exterior**	**…**	**price**
mileage	0.00	0.83	0.83	0.94	0.06	0.95	…	0.99
stock	0.83	0.00	0.07	0.93	0.83	0.95	…	0.98
stock number	0.83	0.07	0.00	0.93	0.83	0.95	…	0.98
color	0.94	0.93	0.93	0.00	0.94	0.11	…	0.92
miles	0.06	0.83	0.83	0.94	0.00	0.95	…	0.99
exterior	0.95	0.95	0.95	0.11	0.95	0.00	…	0.92
…	…	…	…	…	…	…	0.00	…
price	0.99	0.98	0.98	0.92	0.99	0.92	…	0.00

4.2 Quality vs. Workload Size

To find a relationship between the quality and the workload size, we ran the ICD algorithm on randomly sampled fractions of the workload. The experiments showed: the larger workload, the better quality.

5 Conclusion and Future work

We have proposed a novel context-based approach to automatically detecting duplicated attributes in an ontology, which adopts the ICD algorithm from market basket analysis. Our approach has been tested against the real ontology from the car sales domain.

Even though the ICD algorithm appears to converge quickly (typically within 5 iterations) in practice, criteria for that convergence are to be investigated. However, a theoretical analysis of the convergence is difficult, because the ICD algorithm essentially tries to compute fixed points of a non-linear dynamic system. Furthermore,

we'll investigate if some other approaches (such as the term, value and structure-based) can be combined with ours to produce even better results.

Acknowledgments. This work was supported by the Estonian Centre of Excellence in Computer Science (EXCS) funded mainly by the European Regional Development Fund (ERDF).

References

1. Adams, T., Dullea, J., Clark, P., Sripada, S., Barrett, T.: Semantic integration of heterogeneous information. In *CS&I'2000, 5th International Conference on Computer Science and Informatics* (2000)
2. Du, D., LePendu, P.: Ontology-based integration for relational databases. In *SAC'2006, Proceedings of the ACM Symposium on Applied Computing* (2006)
3. Buccella, A., Cechic, A., Brisaboa, N.: An ontology approach to data integration. *Journal of Computer Science and Technology*, Vol. 3, No.2, pp. 62-68 (2003)
4. Cui, Z., Jones, D., O'Brien, P.: Issues in ontology-based information integration. In *IJCAI'2001, 17th International Joint Conference on Artificial Intelligence* (2001)
5. Kruk S., Haslhofer B., Piotrowski P., Westerski A., Woroniecki T.: The role of ontologies in semantic digital libraries. In *NKOS'2006*, *5th European Networked Knowledge Organization Systems Workshop* (2006)
6. Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In *IJCAI'2001, Workshop Ontologies and Information Sharing at 17th International Joint Conference on Artificial Intelligence*, pp. 53-62 (2001)
7. Wu, F., Weld, D.: Automatically refining the Wikipedia infobox ontology. In *WWW'08, 17th International Conference on World Wide Web*. pp. 635-644 (2008)
8. Eyal, A., Gal, A., Jamil, H., Modica, H.: Automatic ontology matching using application semantics. *AI Magazine*, Vol. 26, issue 1, pp. 21-31 (2005)
9. OWL Web Ontology Language Reference, http://www.w3.org/TR/owl-ref (2004)
10. Ehrig, M., Haase, P., Hefke, M., Stojanovic, N.: Similarity for ontologies – a comprehensive framework. In *PAKM'2004, Workshop on Enterprise Modeling and Ontology: Ingredients for Interoperability* (2004)
11. Doan A., Madhavan J., Domingos P., Halevy A.: Learning to map between ontologies on the Semantic Web. In *IWWWC'2002, 11th International World Wide Web Conference*, ACM Press. pp. 662-673 (2002)
12. Berlin J., Motro A.: Autoplex: Automated Discovery of Content for Virtual. *Lecture Notes in Computer Science*, Springer. pp. 108-122 (2001)
13. Madhavan J., Bernstein P., Domingos P., Halevy A.: Representing and reasoning about mappings between domain models. In *AAAI'2002*, *11th National Conference on Artificial Intelligence*. pp. 80-86 (2002)
14. Renda, M., Straccia, U.: Automatic structured query transformation over distributed digital libraries. In *SAC'2006*, *ACM Symposium on Applied Computing*, pp. 1078-1083 (2006)
15. Das, G., Mannila, H.: Context-based similarity measures for categorical databases. In *PKDD'2000, 4th European Conference on Principles of Data Mining and Knowledge Discovery*. pp. 201-210 (2000)
16. Sapia C.: On modeling and predicting query behavior in OLAP systems, In *DMDW'1999*, *International Workshop on Design and Management of Data Warehouses*. pp. 1-10 (1999)
17. Astrova, I.: Toward the Semantic Web – an approach to reverse engineering of relational databases to ontologies. In *ADBIS'05, 9th East-European Conference on Advanced Databases and Information Systems*. pp. 111-122 (2005)

Enabling Content-Based Image Retrieval in Very Large Digital Libraries

Paolo Bolettieri, Andrea Esuli, Fabrizio Falchi, Claudio Lucchese, Raffaele Perego, and Fausto Rabitti

ISTI-CNR, Pisa, Italy.
firstname.lastname@isti.cnr.it

Abstract. Enabling effective and efficient Content-Based Image Retrieval (CBIR) on Very Large Digital Libraries (VLDLs), is today an important research issue. While there exist well-known approaches for information retrieval on textual content for VLDLs, the research for an effective CBIR method that is also able to scale to very large collections is still open. A practical effect of this situation is that most of the image retrieval services currently available for VLDLs are based only on textual metadata. In this paper, we report on our experience in creating a collection of 106 million images, i.e., the CoPhIR collection, the largest currently available to the scientific community for research purposes.We discuss the various issues arising from working with a such large collection and dealing with a complex retrieval model on information-rich features. We present the non-trivial process of image crawling and descriptive feature extraction, using the European EGEE computer GRID. The feature extraction phase is often ignored when discussing the scalability issue while, as we show in this work, it could be one of the toughest issues to be solved in order to make CBIR feasible on VLDLs.

1 Introduction

Everybody knows about the data explosion. According to recent studies, in the next three years, we will create more data than has been produced in all of human history. Regarding images, the Enterprise Strategy Group[1] estimates that more than 80 billion photographs are taken each year. Storing them would require 400 petabytes of storage. Therefore the management of digital images promises to emerge as a major issue in many areas providing a lot of opportunities in the next years, particularly since a large portion of pictures still remains as "unstructured data", i.e., with no meaningful associated tags.

Current searching engines headed by Google are in the center of current information age; Google answers daily more than 200 million queries against over 30 billion items. However, the search power of these engines is typically limited to text and its similarity. Since less than 1% of the Web data is in textual form, the rest being of multimedia/streaming nature, we need to extend our next-generation search to accommodate these heterogeneous media.

[1] http://www.enterprisestrategygroup.com/

A typical approach adopted by the current engines to provide search functionality on these data types is to leverage on the textual information that is potentially associated with the non-textual content, e.g., the text nearby the image published in a Web page, or other attributes, e.g., metadata fields reporting the name of the author, the title, or a list of tags. This approach is adopted by almost all the Web scale engine (e.g., Google, Yahoo, Ask) but also can be found in the latest VLDL initiatives, such as Europeana[2].

An orthogonal approach is the Content-based Image Retrieval (CBIR). It is not a new area as demonstrated by a recent survey [1] which reports on nearly 300 systems, most of them exemplified by prototype implementations. However, the typical database size is in the order of thousands of images. Very recent publicly-available systems, such as ImBrowse, Tiltomo or Alipr[3], declare to index hundreds of thousands of images. There is a huge discrepancy between these numbers and the volumes of images available on current Web, so we decided to investigate the situation by shifting the current bounds up by two orders of magnitude. The result of this process is the CoPhIR (**Co**ntent-based **Ph**oto **I**mage **R**etrieval) collection [2], a collection of 106 million images crawled from the Flickr[4] photo sharing website. The size of the CoPhIR collection goes very far beyond the current practice.

Each image has associated a large number of metadata fields, from the title, description and comments, to the date and the geographical coordinates of the location where the image has been acquired. The availability of a rich amount of metadata information is a nice feature of the collection, that allows to use it to build information-intensive search systems, which is a typical situation in digital libraries. We have enriched the information associated to each image by extracting five MPEG-7 visual features, each one describing some specific characteristic of the image content.

As we detail in Section 2 the image crawling and feature extraction process resulted to be computationally very expensive: the entire process would have required about 12 years on a standard PC. We have used the European EGEE computer GRID to reduce this processing time to just a few days, showing how distributed computing is a crucial element to enable CBIR on the large scale. The CoPhIR collection has been made freely available to the scientific community, and it has been already used to develop and test many CBIR systems.

A relevant result of this experience is that we have observed how the feature extraction process, which is often ignored when discussing the scalability issues of CBIR systems, is probably the toughest issue to be solved in order make CBIR, and in general content-based retrieval for non-textual data, feasible on VLDLs.

---

[2] http://www.europeana.eu/

[3] http://media-vibrance.itn.liu.se/, http://www.tiltomo.com/, http://www.alipr.com/

[4] http://www.flickr.com/

2 Building the Image Collection

Collecting a large amount of images for investigating CBIR issues is not an easy task, at least from a technological point of view. The challenge is mainly related to the size of the collection we are interested in. Shifting the current state-of-the-art bounds of two orders of magnitude means building a 100 million images collection, and this size makes very complex to manage every practical aspect of the gathering process.

2.1 Choosing the Data Source

Given our need of high-quality data, we decided to crawl one of the popular photo sharing sites born in the last years with the goal of providing permanent and centralized access to user-provided photos. This approach has several advantages:
Image Quality. Photo sharing sites mainly store high-quality photographic images.
Collection Stability. These sites provide quite static, long term and reliable image repositories.
Legal Issues Storing for a long time a publicly available image may in some case violate author's copyrights. Since Photo sharing sites are fairly static, we can build a quite stable collection without storing the original files, but maintaining only the hyperlinks to the original photos.
Rich Metadata Photo sharing sites provide a significant amount of additional metadata about the photos hosted, as described in Section 2.2. The availability of rich metadata is a relevant aspect with respect to experimentation, because it allow to investigate on new models of combination of traditional, metadata-based search with content-based search.

Among the most popular photo sharing sites, we chose to crawl Flickr, since it is one with the richest additional metadata and provides an efficient API[5] to access its content at various levels.

2.2 Crawling the Flickr Contents

In February 2007, we crawled the graph of Flickr users, selecting about one million of users. We then exploited the Flickr API to get the whole list of public photos owned by each of these users, retrieving 300 million distinct photo IDs.

For each photo we decided to retrieve almost all information available: title and description, identification and location of the author, user-provided tags, comments of other users, GPS coordinates, notes related to portions of the photo, number of times it was viewed, number of users who added the photo to their favorites, upload date, and all the information stored in the EXIF header of the image file. In order to support content based search, we extracted several MPEG-7 *visual descriptors* (VDs) from each image [3]. A VD characterizes a particular visual aspect of the image. They can be, therefore, used to identify images which have a similar appearance. VDs are represented as vectors, and

[5] http://www.flickr.com/services/api/

the MPEG-7 group proposed a distance measure for each descriptor to evaluate the similarity of two objects [4]. We have chosen five MPEG-7 VDs: Scalable Color, Color Structure, Color Layout, Edge Histogram, Homogeneous Texture.

The extraction of MPEG-7 visual descriptors from high-quality images is computationally very expensive. The MPEG-7 eXperimentation Model (MPEG-7 XM) software running on a AMD Athlon XP 2000+ box takes about 4 seconds to extract the five features from an image of size 500×333 pixels. We can estimate that a single standard PC would need about 12 years to process a collection of 100 million images. It was thus clear that we needed a large number of machines working in parallel to achieve our target collection of 100 million images in a reasonable amount of time.

We developed an application that allows to process images in parallel on an arbitrary (and dynamic) set of machines. This application is composed of three main components: the *image-id server*, the *crawling agents*, and the *repository manager*. The image-id server provide crawling agents with a number of photo identifiers to be processed. The crawling agent asks the image-id server for a set of image identifiers to process. The agent downloads a medium resolution version of the image (500 pixel on the larger side) and all the metadata information associated to it. Then the MPEG-7 XM software is used to extract the aforementioned five VDs. The extracted features and all the available metadata are used to produce an XML file containing the knowledge about the image. In fact, each entry of the CoPhIR collection is an XML structure containing: (i) identification information that allows to link and retrieve the corresponding image on the Flickr Web site; (ii) the image textual data and metadata: author, title, description, GPS location, tags, comments, view count, etc.; (iii) an XML sub-structure containing the information related to the five MPEG-7 visual descriptors.

Given that crawling agent is the most computational-demanding component of our application, we have considered GRID to be the right technology to obtain large amount of computing power we needed, as described in Section 2.3.

The repository manager runs on a large file-server machine providing 10TB of RAID storage. It receives and stores the processed data by the crawling agents.

2.3 Using the GRID for Crawling and Feature Extraction

GRID is a very dynamic environment that allows to transparently run a given application on a large set of machines. In particular, we had the possibility to access the EGEE (Enabling Grids for E-sciencE) European GRID infrastructure[6] provided to us by the DILIGENT (Digital Library Infrastructure on Grid Enabled Technology) IST project[7]. We were allowed to use 35 machines spread across Europe. We did not have an exclusive access to these machines and they were not available all the time. Both hardware and software configurations were heterogeneous.

[6] http://www.eu-egee.org/
[7] http://www.diligentproject.org/

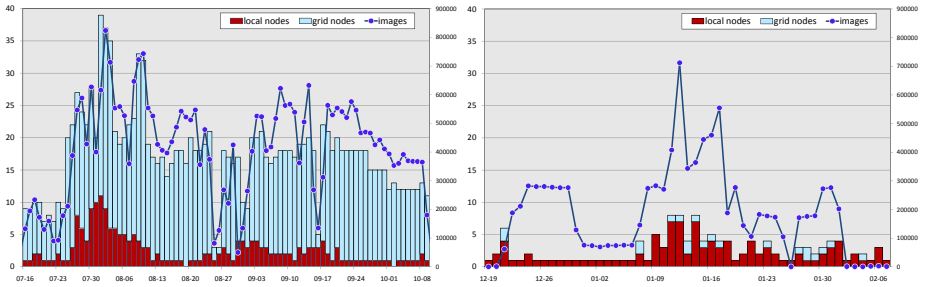


Fig. 1. Number of GRID and local machines available during the two crawling periods: from July 16^{th} to October 9^{th} 2007 (left) and from December 19^{th} 2007 to February 7^{th} 2008 (right).

The crawling agent is logically divided into two modules. The first one manages the communication with the image-id server,uses Flickr APIs, and sends the results to the repository manager. Due to the latencies of the crawling task, the crawling agent can instantiate a number of threads, each of them taking care of processing a different image. The settings which proved well is to have four threads per agent and to process a set of 1,000 images. These parameters induced computations times of 20 to 60 minutes depending on the CPU speed. The second module of the crawling agent is the feature extraction component based MPEG-7 XM software.

Submitting a job to a GRID infrastructure, the user does not have a full control on the time and location where the job runs. The GRID middleware software accepts the job description and schedules it on the next available machine according to internal policies related to the load of each node, the priority of the different organization using the GRID infrastructure, etc. The GRID provides a best-effort service, meaning that a job submitted to the GRID may be rejected and never executed. Indeed, there are several factors that may cause the failure of a job submission. Out of the 66,440 jobs submitted, only 44,333 were successfully executed that means that 33,3 % of the jobs failed for GRID resources unavailability.

Our straightforward approach together with the self-scheduling of images by each crawling agent has two important advantages. First, in case the GRID middleware is not able to deploy the given job, there would be no consequences in the remainder of the system, especially, no image will be skipped. Second, in case of a software update, it is just needed to replace the old version on the repository manager with the new one.

The crawling process took place in two separate periods, both because of GRID availability and because we needed to consolidate the data after the first period. In Figure 1, we report on the number of machines available during the crawling process. During the first period, the GRID provided an average of 14.7 machines out of the 35 and, simultaneously, there were 2.5 local machines available, on average. Also the availability of the machines during the day was unstable: the local machines were mainly available over night while some of the GRID machines were available only for a few hours per day.

2.4 The CoPhIR Collection

The result of the complex crawling and image processing activity described above is the CoPhIR collection. The data collected so far represents the world largest multimedia metadata collection available for research purposes, containing visual and textual information regarding 106 millions images. Given the effort required in building a such large test collection, and the potential interest to the international research community, in order to make experiments in large-scale CBIR, we decided to make it available outside the SAPIR project scope.

The disk space requirement for the CoPhIR collection consist of 245.3 GB for the XML data, 54.14 GB for the image content index, and 355.5 GB for the image thumbnails. CoPhIR images come from 408,889 distinct authors, with a top contributor of 156,344 images (user *conrado4*), and a *median* value of images per author equal to 69. The total number of comments in the collection is 55,188,775. The total number of tag instances is 334,254,683, from a set of 4,666,256 distinct tags. Each image is thus associated on average with 0.52 comments and 5.02 tags.

In the collection, 66,532,213 images (62.77% of the whole CoPhIR) have *popularity* information, i.e., number of views and number of selections as *favorite*. The average number of views per image is 41.7, with a top value of 599,584 views. Just half of the images with popularity information have registered more than 2 views (32,723,369 images, 49.18%), and only 4,963,257 images (7.46% of the part of CoPhIR with popularity information) have been marked has favorite by at least one Flickr user. 8,655,289 images (8.17% of the whole CoPhIR) have *geolocation* information associated to them.

3 Cases of use of the CoPhIR collection

In this section we report some information about research projects and indexing techniques which were able to index the whole CoPhIR collection (i.e., 106 M images) for content-based searching. All of them are using a single metric function defined as a weighted sum of the individual feature distances suggested by the MPEG group. The weights have been reported in [5] and have been defined following the studios reported in [6, 7].

3.1 SAPIR

The scalability challenge is the focus of the European project SAPIR (Search on Audio-visual content using Peer-to-peer Information Retrieval)[8] that aims at finding new ways to analyze, index, and retrieve the tremendous amounts of speech, image, video, and music that are filling our digital universe. CoPhIR has been built and used in SAPIR for experimenting various approach: text based, similarity search, geographic search, etc. All P2P index and search techniques [8] are based on the metric space model and implemented over the same framework - the Metric Similarity Search Implementation Framework (MESSIF) [9].

[8] SAPIR European Project, IST FP6: http://www.sapir.eu/

For the scope of improving throughput and response time, during the SAPIR project a metric cache was developed [10]. Unlike traditional caching systems, the proposed a caching system might return a result set also when the submitted query object was never seen in the past. In fact, the metric distance between the current and the cached objects is used to drive cache lookup, and to return a set of approximate results when some guarantee on their quality can be given.

3.2 Metric Inverted File

Amato and Savino [11] have proposed the *Metric Inverted File* (MIF), a novel method for approximate similarity search based on the use of *permutations* to represent the indexed objects. A permutation, assigned to each indexed object, consists of a list of the elements of a set of *reference object*, sorted by their order of similarity with the object. The MIF builds an inverted-list-based data structure on permutations, which allows to approximate the exact order of similarity of the indexed objects, with respect to a query and the similarity measure in use, by efficiently computing a similarity measure on permutations. Although the authors report in [11] results of experiments on a smaller collection, a demo of MIF on the CoPhIR collection is available[9], showing a good trade-off on efficiency/effectiveness in the retrieval process.

3.3 MiPai

MiPai[10] [12] is a CBIR system based on the PP-Index [13] data structure for approximated similarity search. The PP-Index is an efficiency-aimed data structure that belongs to the family of the permutation-based indexes [11]. It has nice parallelization properties that allow to tune the efficiency/effectiveness trade-off with respect to the available hardware resources. Experiments on the CoPhIR collection have shown that the PP-Index is able to achieve almost exact results (i.e., 97% precision with respect to the exact order defined on the similarity measure in use) within just a few seconds, while an exaustive search process would take more than an hour. Moreover, it is interesting to note that it took just 12.5 hours to build a PP-Index on the whole CoPhIR collection on a single PC, showing how the phase of index creation from the extracted feature can be orders of magnitude smaller than the preceding phase of features extraction.

4 Conclusions

No doubts that the scalability issue for new digital data types is a real issue, which can be nicely illustrated by difficulties with the management of the fast growing digital image collections. In this paper, we focus on a strictly related challenge of scalability: to obtain a non-trivial collection of images with the corresponding descriptive features.

We have crawled a collection of over 100 million high-quality digital images, which is almost two orders of magnitude larger in size than existing image

[9] http://mi-file.isti.cnr.it/CophirSearch/
[10] http://mipai.esuli.it/

databases used for content-base retrieval and analysis. Using a GRID technology, we have extracted five descriptive features for each image. This information is kept handy in XML files – one for each image – together with the metadata and links to original images in Flicker. This unique collection is open to the research community for experiments and comparisons. More than 50 research institution worldwide already asked access to the CoPhIR collection by registering at the CoPhIR Web site[11], and by signing the CoPhIR Access Agreement, which establishes conditions and terms of use for the collection.

References

1. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys (2008) To appear.
2. Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., Rabitti, F.: CoPhIR: a test collection for content-based image retrieval. CoRR abs/0905.4627 (2009)
3. ISO/IEC: Information technology - Multimedia content description interfaces. (2002) 15938.
4. Manjunath, B., Salembier, P., Sikora, T., eds.: Introduction to MPEG-7: Multimedia Content Description Interface. J. Wiley & Sons, New York, USA (2002)
5. Batko, M., Falchi, F., Lucchese, C., Novak, D., Perego, R., Rabitti, F., Sedmidubsky, J., Zezula, P.: Building a web-scale image similarity search system. Multimedia Tools and Application (2009) forthcoming.
6. Amato, G., Falchi, F., Gennaro, C., Rabitti, F., Savino, P., Stanchev, P.: Improving image similarity search effectiveness in a multimedia content management system. In: Proc. of Workshop on Multimedia Information System (MIS). (2004) 139–146
7. Stanchev, P., Amato, G., Falchi, F., Gennaro, C., Rabitti, F., Savino, P.: Selection of mpeg-7 image features for improving image similarity search on specific data sets. In: Proceedings of the 7-th IASTED International Conference on Computer Graphics and Imaging (CGIM 2004), ACTA Press (August 2004) 395–400
8. Batko, M., Novak, D., Falchi, F., Zezula, P.: Scalability comparison of peer-to-peer similarity search structures. Future Generation Computer Systems **24**(8) (2008) 834 – 848
9. Batko, M., Novak, D., Zezula, P.: MESSIF: Metric similarity search implementation framework. In: Proc. of DELOS Conference. Volume 4877 of LNCS. (2007) 1–10
10. Falchi, F., Lucchese, C., Orlando, S., Perego, R., Rabitti, F.: Caching content-based queries for robust and efficient image retrieval. In: EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology, ACM (2009) 780–790
11. Amato, G., Savino, P.: Approximate similarity search in metric spaces using inverted files. In: INFOSCALE '08: Proceeding of the 3rd International ICST Conference on Scalable Information Systems, Vico Equense, Italy (2008) 1–10
12. Esuli, A.: MiPai: using the PP-Index to build an efficient and scalable similarity search system. In: SISAP '09, Proceedings of the 2nd International Workshop on Similarity Search and Applications, Prague, CZ (2009) 146–148
13. Esuli, A.: PP-index: Using permutation prefixes for efficient and scalable approximate similarity search. In: 7th Workshop on Large-Scale Distributed Systems for Information Retrieval (LSDS-IR'09). (2009) 17–24

[11] http://cophir.isti.cnr.it

Semantic Journal Mapping for Search Visualization in a Large Scale Article Digital Library

Glen Newton[1,2] and Alison Callahan[1] and Michel Dumontier[2,3,4]

[1] National Research Council Canada, Ottawa, Canada
[2] Department of Biology,
[3] Institute of Biochemistry,
[4] School of Computer Science, Carleton University, Ottawa, Canada

Abstract. In this paper we examine the scalability and utility of semantically mapping (visualizing) journals in a large scale (5.7+ million) science, technology and medical article digital library. This work is part of a larger research effort to evaluate semantic journal and article mapping for search query results refinement and visual contextualization in a large scale digital library. In this work the Semantic Vectors software package is parallelized and evaluated to create semantic distances between 2365 journals, from the sum of their full-text. This is used to create a journal semantic map whose production does scale and whose results are comparable to other maps of the scientific literature.

1 Introduction

The goal of this work is to create semantic journal maps to support the user search experience in a large scale digital library of science, technology and medical (STM) journal articles. By projecting article search results onto a semantic map, we seek to visualize and contextualize the query results, and offer interactive tools for users to refine queries and discover relevant articles.

Before article query result mapping can occur, we need to find and validate a technique that can scale up to tens of millions of terms and yet generate understandable and useful semantic journal mappings.

Visual representations of the scientific literature — often called *Maps of Science* — have been studied extensively over the last decade[20,1,3,5]. These maps have been constructed for a number of reasons: for informing science policy[16,14]; comparing author groupings within a discipline[6]; examining within discipline journal influence[2]; comparing the innovation systems between and within countries[17,9]; visualizing and discovering research fronts[12] etc. The majority of these maps have been constructed using citation, author, or term analyses, with only a minority using content-based methods like latent semantic analysis (LSA)[8,7]. This is due to the scalability issues of LSA, whose solution depends on singular value decomposition which becomes untenable when tens of millions of terms and/or items are involved.

Semantic Vectors (SV) [21] method produces similar results to LSA but uses a much less computationally expensive method based on random vectors[4]. This paper evaluates the scalability and quality of journal semantic mappings generated by SV.

Table 1. Category Statistics

Category	**# Journals per category**	**Category**	**# Journals per category**
Agricultural and Biological Sciences	358	Arts and Humanities	70
Biochemistry, Genetics, and Molecular Biology	240	Business, Management and Accounting	106
Chemical Engineering	126	Chemistry	226
Civil Engineering	64	Computer Science	218
Decision Sciences	50	Earth and Planetary Sciences	146
Economics, Econometrics and Finance	112	Energy and Power	73
Engineering and Technology	328	Environmental Science	138
Immunology and Microbiology	104	Materials Science	160
Mathematics	205	Medicine	671
Neuroscience	103	Pharmacology, Toxicology and Pharmaceutics	73
Physics and Astronomy	210	Psychology	126
Social Science	222		

2 Data and Methods

The data is derived from NRC-CISTI's onsite collection of full-text STM scientific articles. This dataset is approximately 4100 journals and 8.4 million articles, and primarily covers the publishing years 1995 to 2009. The collection includes metadata and full-text for all articles. All journals are classified into one or more of 23 categories by library cataloguers (Table 1).

2.1 Data Preparation

Articles were restricted to those whose metadata indicated they had authors and abstracts, as a filter to reduce the number of notices, obituaries, etc. in the collection. As the text extraction software is language specific, non-English language journals were also filtered out. In addition, journals with less than 50 articles in total were excluded. The resulting collection amounted to 5,733,721 articles. Table 1 shows the journals per category for this collection. The average number of categories per journal for this collection was 1.53, comparable to the

Fig. 1. Two-dimensional Projection of Semantic Mapping of all 2365 journals. Each color represents a category.

1.6 reported in [10] in their journal citation analysis and mapping based on ISI Science Citation Index.

Using the LuSql[5] Lucene tool, the full-text (only) of the articles from each journal was concatenated into a single (rather large) document. This *journal* document (one per journal) was then indexed using Lucene[6] v2.4, using an aggressive stopword list and the Porter stemmer (the Snowball analyzer). Semantic Vectors[7] v1.18 BuildIndex was applied to the Lucene index, configured to use 512 dimensions.

A parallelized version of the SV was used to query the resulting model, using each document in the index as a query, resulting in a distance matrix between all documents (journals). Using the R statistics software[8], this distance matrix was scaled to two dimensions using classical multidimensional scaling (MDS) with the `cmdscale()` function. An interactive application[9] was constructed to visualize and validate the two dimensional projection using the Processing[19] language.

[5] `http://sourceforge.net/projects/lusql/`
[6] `http://lucene.apache.org/`
[7] `http://code.google.com/p/semanticvectors`
[8] `http://www.r-project.org/`
[9] `http://cuvier.cisti.nrc.ca/~gnewton/torngat/applet/index.html`

3 Results and Discussion

All computation performed on a Dell PowerEdge 1955 Blade server, 2 x dual-core Xeon 5050 processors with 2x2MB cache, 3.0 Ghz 64bit, 32GB RAM, attached to a Dell EMC AX150 storage arrays via SilkWorm 200E Series 16-Port Capable 4Gb Fabric Switch. The operating system was Linux openSUSE 10.2 (64-bit X86-64), kernel 2.6.18.8-0.10-default #1 SMP, Java version 1.6.0.07 (build 1.6.0 07-b06) Java HotSpot 64-Bit Server VM (build 10.0-b23, mixed mode).

3.1 Scalability

The Lucene index produced by indexing the journal articles was 43GB in size, and took 13 hours and 51 minutes to produce. This index contained 21.6 million terms. Semantic Vectors indexing took 58 minutes to produce a document vector index of 885MB. The parallelized SV document querying took six minutes to complete the document-document distance comparisons.

3.2 Semantic Mapping

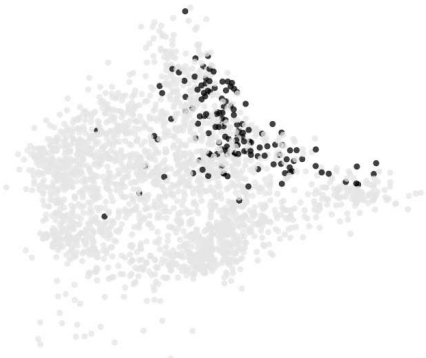
Figure 1 shows the 2-D MDS of the semantic distance space for all journals, colored by the categories defined in the collection metadata. Figures 2, 3 and 4 are groupings of the categories into the *harder* sciences and engineering, agricultural and biomedical, and inter-disciplinary and non-science, respectively. The sub-diagrams show each category within these clusters. If we use the geographic direction idiom, Figure 2 shows a north-east to south-east progression from Chemistry, Fig. 2(a), through Material Sciences, Fig. 2(b), to Physics and Astronomy, Fig. 2(c), to Engineering and Technology, Fig. 2(d), on to Mathematics, Fig. 2(e), then swinging west into Computer Science, Fig. 2(f). Chemical Engineering and Civil Engineering — somewhat under-represented in this study — are more central-north, Fig. 2(g), and central, Fig. 2(h), respectively.

The biomedical cluster forms most of the western portion of the map (Fig. 3) and seems to have greater cluster overlap than the "hard sciences". North-central and bridging to Chemistry in the north-east is Biochemistry, Genetics and Molecular Biology, Fig. 3(a), Agriculture and Biological Sciences, Fig. 3(b). Pharmacology, Fig. 3(c) and Immunology and Microbiology, Fig. 3(d), form clusters in the north-west, with Neuroscience, Fig. 3(e), in the west. Medicine, Fig. 3(f), occupies much of the south-west and is a larger cluster, overlapping with all other biomedical categories as well as many of the other categories. Psychology, Fig. 3(g), occupies the south-west, with significant overlap with Medicine and Neuroscience.

The third cluster is a catch-all of inter-disciplinary science and non-STM journals. Environmental Science, Fig. 4(a), and Earth and Planetary Sciences, Fig. 4(b), are both central, surrounded by all other clusters, an indication of their complex relationships with other disciplines, both STM and non-STM. Energy and Power Fig. 4(c), has a similar positioning, although it is an even more vague category. Decision Science, Fig. 4(d), — another under–represented category —

(a) Chemistry

(b) Materials Science



(c) Physics and Astronomy

(d) Engineering and Technology

(e) Mathematics

(f) Computer Science

(g) Chemical Engineering

(h) Civil Engineering

Fig. 2. *Harder* sciences and engineering categories

(a) Biochemistry, Genetics and Molecular Biology

(b) Agriculture and Biological Sciences

(c) Pharmacology

(d) Immunology and Microbiology

(e) Neuroscience

(f) Medicine

(g) Psychology

Fig. 3. Agriculture and bio-medical categories

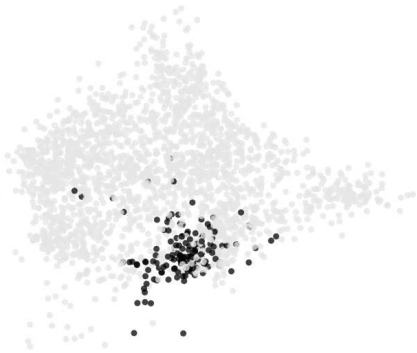
(a) Environmental Science

(b) Earth and Planetary Sciences

(c) Energy and Power

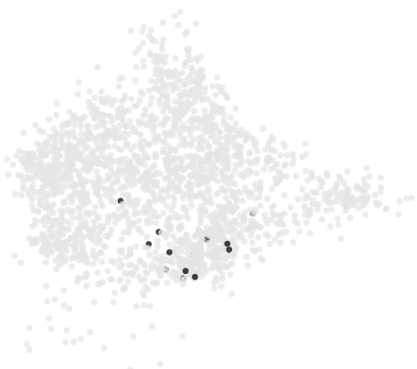
(d) Decision Science

(e) Economics



(f) Social Sciences

(g) Business, Management and Accounting



(h) Arts and Humanities

Fig. 4. Inter-disciplinary and non-science categories

is a rather spread-out cluster and overlaps with Economics, Computer Science and Mathematics. Economics, Fig. 4(e), is central south, and forms a bridge between Mathematics to the east, and Social Sciences and Business, Management and Accounting to the west. Social Science, Fig. 4(f), is central south, along with Business, Management and Accounting, Fig. 4(g), and for the most part, the also under-represented Arts and Humanities, Fig. 4(h).

This mapping is very similar to that reported by other efforts[3,13], and particularly to that described in [10], where the the authors identify a *medical pole* and *hard-science pole*: "*Chemistry plays a brokerage role between Physics and Material Sciences, on the one side, and core Biomedical Sciences such as Biophysics and Biochemistry, on the other.*" ([10, p.355],)

4 Conclusion

Semantic Vectors produced a robust semantic mapping that was comparable to other maps made with other techniques. In this case, however, no structured information such as keywords, authors or citations were needed to produce these results. SV was able to scale to the extremely large number of terms in this digital library.

4.1 Future Work

With the scalability and semantic usefulness issues resolved, the next step in this work is to project search results, i.e. the first 100–200 articles resulting from a query, onto the journal semantic space. The underlying journal space contextualizes the user's query, and we envision the ability to zoom in local to the bounding polygon of the search results. This interface would allow a user to visualize semantically similar articles, as well as identifying important journals related to their query. Three-dimensional MDS scaling and visualization will be explored in the hope of better discrimination and less loss of structure over the two-dimensional MDS used here. We will be evaluating non-metric MDS to see if it generates better results than the classical MDS used in this work.

We are also interested in evaluating the finer–grained attributes of SV-generated semantic maps, with an interest in creating domain–specific maps for queries that fall within a particular domain.

5 Acknowledgements

The authors would like to thank NRC-CISTI colleagues Greg Kresko for technical support and Andre Vellino and Jeff Demaine for their input on this project.

References

1. Borner, K., Chen, C., Boyack, K. Visualizing Knowledge Domains. In Blaise Cronin (Ed) Annual Review of Information Science & Technology volume 37 (2003).

2. van Campenhout, C., van Caneghem, T., van Uytbergen, S. A comparison of overall and sub-area journal influence: The case of the accounting literature. Scientometrics 77(1) 61–90 (2008)
3. Boyack, K., Klavans, R., Borner, K. Mapping the backbone of science. Scientometrics 64(3) 351–374 (2005)
4. Kanerva, P. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. Cogn Comput 1 139-159 (2009)
5. Klavans, R., Boyack, K. Toward a consensus map of science. JASIST 60, 455–476 (2009)
6. Kreuzman, H. A co-citation analysis of representative authors in philosophy: Examining the relationship between epistemologists and philosophers of science. Scientometrics 50(3) 525–539 (2001)
7. Landauer, T., Laham, D., Derr, M. From paragraph to graph: Latent semantic analysis for information visualization. PNAS 101 S1 5214–5219 (2004)
8. Landauer, T., Dumais, S. A solution to Plato's problem: The latent semantic analysis theory of acquisition. Psych Rev 104(2) 211–240 (1997)
9. Leydesdorff, L., Yuan, S. National and international dimensions of the Triple Helix in Japan: University-industry-government versus international coauthorship relations. JASIST 60(4) 778–788 (2009)
10. Leydesdorff, L., Rafois, I. A Global Map of Science Based on the ISI Subject Categories. JASIST 60(2), 348–362 (2009)
11. Michalek, R., Tarantello, G. Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems. J. Diff. Eq. 72, 28–55 (1988)
12. Miguel, S., Moya-Anegn, F., Herrero-Solana, V. A new approach to institutional domain analysis: Multilevel research fronts structure. Scientometrics 74(3) 331–344 (2008)
13. Moya-Anegón, F., Vargas-Quesada, B., Chinchilla-Rodríguez, Z., Corera-Álvarez, E., Munoz-Fernández, F., Herrero-Solana, V. Visualizing the marrow of science. JASIST 58(14) 2167–2179
14. Noyons, E., Calero-Medina, C. Applying bibliometric mapping in a high level science policy context: Mapping the research areas of three Dutch Universities of Technology. Scientometrics 79(2) 261–275 (2009)
15. Newton, G. LuSql: (Quickly and easily) Getting your data from your DBMS into Lucene. Code4Lib Conference, Providence, RI (2009)
16. Noyons, E. Science Maps Within a Science Policy Context. In H. Moed (Ed) Handbook of Quantitative Science and Technology Research, 237-255. Kluwer (2004)
17. A comparison of the knowledge-based innovation systems in the economies of South Korea and the Netherlands using Triple Helix indicators. Scientometrics 65(1) 3–27 (2005)
18. Rabinowitz, P. On subharmonic solutions of a Hamiltonian system. Comm. Pure Appl. Math. 33, 609–633 (1980)
19. Reas, C., Fry, B. Processing: A programming handbook for Visual Designers and Artists. MIT Press (2007)
20. Small, H. A general framework for creating large-scale maps of science in two or three dimensions: The SciViz system. Scientometrics 41(1-2) 125–133 (1998)
21. Widdows, D., Ferraro, K. Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08) (2008)

Exploiting individual users and user groups interaction features: methodology and infrastructure design

Emanuele Di Buccio and Massimo Melucci

Department of Information Engineering, University of Padua, Italy
{dibuccio,melo}@dei.unipd.it

Abstract. The user may be a source of evidence for supporting information access through Digital Library (DL) systems. In particular, the features gathered while monitoring the interaction between the user and a DL system can be used as implicit indicators of the user interests. However, each user has his own style of interaction and a feature which is a reliable indicator with regard to one user may be no longer reliable when referred to another user. This suggests the need to develop personalized approaches for each user which are tailored for each search task. Nevertheless, the behavior of a group of interrelated users, e.g. performing the same task, may improve the contribution provided by the personal behavior; for instance, some interaction features, if considered individually, are more reliable with regard to a group of users. This paper introduces a methodology for exploiting both the behavior of individual users and group of users as sources of evidence. The paper also introduces a software infrastructure implementing the methodology. The methodology is mainly based on a geometric framework while the software infrastructure is based on a partially decentralized Peer-To-Peer (P2P) network, thus permitting the management of different sources of evidence.

1 Introduction

The heterogeneity and the size of the data made available by Digital Library (DL) systems, the high number of users and user groups accessing such systems, and the variety of the information needs require strategies which help the users to access relevant information. This paper addresses the design of strategies for information access both at methodological and infrastructural level. First, the paper introduces a methodology for exploiting a variety of sources of evidence for enhancing information access. The specific sources considered in this paper are the features gathered by monitoring the behavior of a user or a group of users somehow interrelated; Section 2 provides motivations for this choice. The methodology is structured in four steps, the aim of which is to support the selection of the sources, to collect the evidence, to model the sources and their relationships in order to make them usable, and to assist their use for accessing relevant information. Sources and relationships are modeled and then exploited through a previously proposed geometric framework, which provides a uniform

and usable representation for them in terms of vector space subspaces. A detailed description of the methodology will be provided in Section 4 and Section 5. This paper also provides a contribution at the infrastructural level as the methodology is complemented by the design of a partially decentralized infrastructure based on the P2P paradigm which allows data about the behavior of the individual users and users groups to be managed. Section 3 provides motivations for the infrastructure. Section 5 explains its role in regard to the proposed methodology.

2 User behavior: a source to support information access

Information about user behavior when interacting with DL systems can be a useful source of evidence to improve the systems capability of providing the relevant documents which the user is looking for. The user can explicitly be asked to provide information about his intents or interests. For instance, the user can explicitly rate images, audio files, videos, or label documents with tags to describe them. This information is a valuable source for building user profiles which can be adopted to suggest other possible results of interest. Explicit feedback has been shown to be an effective strategy to support the user during searching, e.g. to refine or expand queries.

Despite its effectiveness, explicit feedback is not always perceived useful by users due to the effort required by direct involvement of the user. An alternative is the adoption of implicit indicators as a source of evidence for relevance feedback. The expression "implicit indicators" refers to those features collected without direct involvement by the user; examples include user interaction features, e.g. display-time, click-through data or the features gathered during the user study described in [1]. Implicit Relevance Feedback (IRF) algorithms [2] exploit implicit features to provide feedback to the user, thus preserving the benefits of explicit relevance feedback and removing its burdens.

When user behavior is used as a source of evidence for IRF algorithms, the behavior features can be collected with regard to the individual user or group of interrelated users — e.g. a group of users who perform the same task, submit the same topic or belong to a social network. When the individual user is adopted as a source of evidence, the features have *personal granularity*[1], e.g. the display-times or the click-through data are gathered only by the observation of the considered user. In the remainder of this paper we will refer to the set of personal granularity features, or to the result of their arrangement or combination, as the *personal behavior dimension*. The approach proposed in [3], for instance, simultaneously exploits multiple interaction features with personal granularity for developing enhanced implicit feedback models personalized for each user and tailored for each search task. One of the issues when exploiting personal information is privacy concerns – if a centralized system is adopted, the user may be reluctant to use personalization if it requires providing personal data.

As mentioned above, not only the individual user, but also groups of users explicitly or implicitly formed can be used as a source of evidence for IRF algo-

[1] The *granularity* of the features is the level of detail at which the features are observed.

rithms – similar to the personal features, the set of behavior features referring to the group can be named *group behavior dimension*. One of the reasons for exploiting a group of users as a source is, for instance, that some interaction features, e.g. display-time, are not reliable indicators when considered individually and with regard to a single user performing a specific task; whereas multiple users performing the same task tend to interact more consistently [4]. Implicit features at *group granularity* were adopted in [5], where the *hit-matrix* which stores the search behavior data – the element (i, j) of the matrix represents the number of times the page j was accessed with regard to the query i – maintains information at the *community* granularity level – in [5] a "community" represents a set of users working in the same company or accessing an interest-specific portal. Another reason for exploiting group evidence is that the contribution provided by the group can be different from that provided by the individual user – e.g. in [6] the information gathered by groups of users is adopted to enhance personalization.

3 An infrastructure to manage user behavioral features

The collection and the aggregation of user information for enhancing information access may cause concerns about privacy preservation. In [7] the authors proposed several levels of privacy protection in personalized search and then examined several software architecture for personalization with regard to the defined privacy levels. The rationale of the infrastructure proposed in this paper is close to the architecture named *client-server cooperative personalization* in [7]. In this architecture the user information is stored on the client side; at query time the client extracts the information from the user profiles and sends it to the server, namely the search engine, where personalization is performed.

The specific infrastructure adopted in this paper is based on the SPINA software architecture [8]. Unlike other papers, a partially decentralized infrastructure based on the P2P paradigm is proposed in this paper. SPINA was designed and developed for providing indexing and retrieval of unstructured documents distributed across P2P networks which are unstructured (i.e. no DHT-like data structures), hybrid (i.e. the simultaneous presence of peers and ultra-peers) and hierarchical (i.e. each peer refers to one and only one ultra-peer which serves a group of peers acting as a hub/router for queries sent by a peer in its group). Ultra-peers are peers with previously established attributes – for example with more CPU, bandwidth or disk than the others – which are *dynamically* elected from normal peers. An instance of this kind of network is depicted in Figure 1. The retrieval strategy adopted in SPINA exploits indexes at different granularities, particularly with peer and ultra-peer granularity. In these indexes the elements in the posting list associated with each feature are the identifier of the peer (or the ultra-peer) and the weight of the feature in that peer (or ultra-peer). These weights are computed by the aggregation of the statistics of the features according to the level hierarchy.

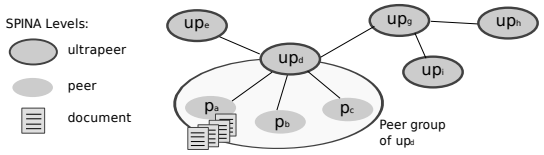


Fig. 1: SPINA Levels.

The functionalities provided by SPINA to manage the features at the two granularity levels can support the methodology proposed in this paper based on the two different behavioral feature granularities, i.e. individual user and group granularity. In particular, the rationale is that features with personal behavior granularity are stored in the peers, while the group behavior granularity features are stored on the ultra-peer side and are obtained as aggregation of the features stored in the peers which refer to that ultra-peer. The ultra-peer acts as a proxy server, thus all the users interacting with peers in the ultra-peer group are interpreted as an individual user.

4 A methodology to exploit user behavior

Information access and retrieval systems can exploit the evidence provided by different sources for improving retrieval effectiveness. Although mostly used metadata, content and structure of the documents are only a subset of the available sources. Instances of other sources of evidence are the search task, the specific topic within the task [1], or the location. The complexity of the design and the development of approaches to exploit different sources is not only due to the number of sources involved, but also to the relationships existing among such sources. Let us consider a user who is looking for restaurants in London and interacts with the results returned by the DL system. If the keyword "jazz" appears in the first result selected by the user – e.g. in the snippet, the title or the url related to the result – the user behavior probably suggests he is more interested in jazz restaurants than in generic ones. Therefore, a relationship can exist between the two sources considered, that is, the user behavior and the terms appearing near the query features in the displayed results.

The point here is that the different sources are not necessarily independent of each other – the features observed from a source (e.g. the behavior) are "entangled" with the features observed from another source (e.g. the particular meaning of a query feature in the selected results). The design of different approaches, one for each source, may fail to deal with the challenge of modeling the relationships existing among sources and consequently fail to exploit them. The methodology introduced in [9] and refined in this work, aims at addressing this problem through a uniform and usable representation of the different sources and their relationships. The methodology is structured in four steps which are depicted in Figure 2 with regard to a specific source, namely user behavior.

The first step of the methodology is the *selection of the sources* adopted to support information access. This choice is not unique, but can depend on the

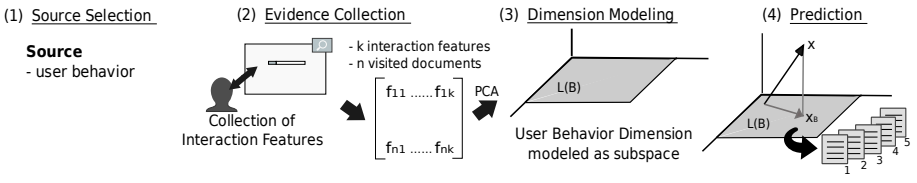


Fig. 2: Methodology Steps for User Behavior Dimension.

specific domain of application: the sources involved when the tool is developed for a mobile device are different from those which can be involved when designing a desktop search tool or when addressing the problem of enterprise search. The specific source considered in this work is the behavior of the user when interacting with the results returned by the system. In accordance with the terminology adopted in Section 2, this source will be named as *user behavior dimension*.

Once the sources have been selected, the dimensions identified to describe such sources have to be modeled. The model of a dimension is built around the evidence collected by monitoring the sources related to this dimension. Feature selection and gathering constitutes a preliminary step to support dimensions modeling: this step will be named *evidence collection*. The selection of the suitable features is part of the design to implement the methodology, since it affects the modeling step. Indeed, a feature or a set of features have to be a reliable indicator of the user needs, intents or interest. Interaction features have been selected in this work to describe the user behavior dimension. Moreover, the feature selection step provides some requirements to the design of the tool which practically will gather the features. For instance, interaction features can be monitored by an extension of a browser or a plug-in for a media-player application.

The collected features constitute a representation of the source, but this representation is not always directly usable: features often appear as rough and noisy data. The purpose of the *dimension modeling* step is to model a source in order to make it usable, thus supporting access to relevant information. In this work the modeling step is addressed by the adoption of the geometric framework proposed in [10], whose rationale is to use vector space subspace as a construct to model a dimension or a set of dimensions. In order to understand the rationale of this step, let us consider the result of the previous step when applied to the user behavior dimension. Let us suppose the user interaction behavior was monitored when accessing and interacting with the first k results visited. The gathered interaction features can be prepared in a matrix A, whose (i, j) element is the feature j observed during the visit of document i — see Step (2) in Figure 2. The matrix A is a vector-based representation of the observed data, but this representation does not always reveal the logical structure underlying the data. A matrix transformation technique can be adopted to reveal the logical structure: for instance in [3] Principal Component Analysis (PCA) of $A^T A$ is adopted to compute the vector space basis. A subset of the eigenvectors computed by PCA can be used as a basis to model the user behavior dimension: the basis spans a subspace which is the model of the dimension – e.g. the subspace $L(B)$ in Fig. 2.

The last step is the *prediction* of the documents which can be relevant to the user information need. This step can be performed by the adoption of the trace-based function adopted in [10], which provides a measure of the degree to which the modeled dimensions occur in a document. Documents can be ranked according to this measure, which has proven to be a probability measure. The rationale is to compute the distance between the vector representation of the documents in terms of features of the selected dimension – e.g. vector $\mathbf{x}$ depicted in step (4) of Figure 2 – and the subspace modeling the dimension(s) spanned by the computed vector space basis – e.g. the subspace $L(B)$ in Figure 2.

5 Implementing user behavior granularities

The methodology introduced in Section 4 is general and in order to apply it to a specific dimension an implementation of the four steps is required. This section aims at providing a possible implementation with regard to the user behavior as a source, moreover focusing on the infrastructure to support this implementation.

Let us consider a user who interacts with a DL node, namely a peer, in order to submit a query. Since in this work interaction features are adopted to model the user behavior dimension, the peer needs a monitoring tool able to gather information about the interaction between the user and the application to access information – e.g. if the application is developed to run in a web browser, this monitoring tool may be a browser extension. When the system returns the results to the user, the monitoring tool gathers interaction features with regard to the first k visited documents and then this evidence is used to obtain a model of the user behavior dimension — a possible approach is the one proposed in [3] and mentioned in the description of the *dimension modeling* step, i.e. modeling a dimension as a vector space subspace. Once the subspace has been computed, the documents can be re-ranked by the trace-based function proposed in [10]. The documents to be ranked can be part of the local collection or distributed over Internet or across a P2P network. If only the local collection is considered, the user behavior can be useful only for the documents already seen: no evidence is available for the other documents, but the user profile can be enriched across different searches.

The methodology together with the infrastructure can provide a different contribution when documents other than those stored locally are accessed. SPINA provides not only functionalities to perform local search, but also to search documents distributed across a P2P network. In particular, a SPINA peer forwards the query to its referring ultra-peer, which according to the query features — e.g. keywords or audio patterns — selects and forwards the query to the most promising peers in its group — details about the algorithm can be found in [8]. The contacted peers locally retrieve the most promising documents, which are returned to the ultra-peer. Moreover, the ultra-peer can forward the query to its most promising neighboring ultra-peers which will iterate the search process in their groups and return the results to the starting ultra-peer.

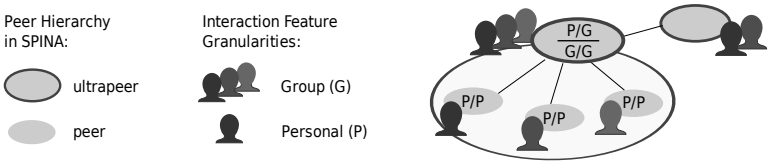


Fig. 3: SPINA to support the user behavior dimension.

Accessing the documents distributed across the network not only augments the contents available, but also improves the support provided by the user behavior dimension. Indeed, the retrieved documents may be accessed by other users, for instance when submitting similar queries or performing similar tasks. Each user can set its peer to periodically provide his behavioral features to its referring ultra-peer. The ultra-peer aggregates the contribution provided by such interaction features with the information already present with regard to the same task, similar topics, or in general according to the selected aggregation policy. For instance, the number of clicks on a specific document with regard to a particular user when performing a certain task can be added to the number of clicks stored in the ultra-peer with regard to the same document and the same task. This information will not refer to the behavior of the individual user, but to the behavior of the group of users interacting with peers in the ultra-peer group. Since the features at user granularity are not maintained in the ultra-peers, the user may be encouraged to provide its contribution in terms of interaction features.

These group granularity features can be used to complement the personal granularity features when the latter are not available for some documents which can be described in terms of group granularity interaction features. In particular, after k documents have been visited, the peer sends the monitored interaction features with personal granularity to the ultra-peer, which builds a representation of the personal behavior dimension and uses the group granularity interaction features available to re-rank the documents against the personal interaction dimension. Finally, the ultra-peer returns the results to the requesting peer and updates the group granularity features with the features previously sent by the peer. This approach can be interpreted as a P/G combination exploiting personal features (P) to model the dimension and the group granularity features (G) to represent the documents. By using the same labeling scheme, the feedback completely performed locally, namely in the peer, can be labeled as P/P which exploits the personal features both to build the dimension model and to represent the documents. Another possible approach is the G/G case which exploits the group granularity features both for dimension modeling and document representation – this approach is exploited at the ultra-peer level. Since each user has his own style of interaction, this combination based on the group behavior can be useful to diversify result suggestions. Figure 3 depicts the level of the peers hierarchy where each of the three different approaches – i.e. P/P, P/G and G/G – are performed.

6 Concluding Remarks

In this work we have proposed a methodology and an infrastructure which aims at supporting information access by exploiting user behavior described in terms of interaction features at different granularities, in particular, referred to individual users (P) and user groups (G). Some preliminary experiments conducted on the effectiveness of the different combinations of interaction sources, showed how the P/P combination is the most effective one, followed by P/G and G/G; we are currently investigating if the different combinations provide diverse contributions and how to combine them.

The main issue under investigation is the grouping criterion, since it affects the policy of aggregation of the personal interaction features in order to obtain the features with group granularity. A possible solution is grouping users by tasks [4, 6]; in our preliminary experiments the users were grouped according to the search task, but groups were manually defined. The issue to be addressed is how to automatically create groups; a solution to this issue may also suggest policies according to which the peers join peer groups, namely select ultra-peers. In regard to the current implementation of SPINA, behavioral features can be a source of evidence for supporting the result merging procedure. A result merging technique is required because the scores assigned to the documents according to their contents by the different peers are not directly comparable. Behavioral features can be adopted to re-rank the results in the ultra-peers and to present a final ranked list of results to the user.

References

1. D. Kelly. Measuring online information seeking context, part 1: Background and method. *JASIST*, 57(13):1729–1739, 2006.
2. D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
3. M. Melucci and R.W. White. Utilizing a geometry of context for enhanced implicit feedback. In *Proceedings of CIKM'07*, pages 273–282, 2007.
4. R.W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of CIKM'06*, pages 297–306, New York, NY, USA, 2006. ACM.
5. B. Smyth. A community-based approach to personalizing web search. *Computer*, 40(8):42–50, 2007.
6. J. Teevan, M. Ringel Morris, and S. Bush. Discovering and using groups to improve personalized search. In *Proceedings of WSDM'09*, Barcelona, Spain, 2009.
7. X. Shen, B. Tan, and C. Zhai. Privacy protection in personalized search. *SIGIR Forum*, 41(1):4–17, 2007.
8. E. Di Buccio, N. Ferro, and M. Melucci. Content-based Information Retrieval in SPINA. In *Proceedings of IRCDL2009*, Padova, Italy, 2008.
9. E. Di Buccio and M. Melucci. Towards a Methodology for Contextual Information Retrieval. In *Proceedings of the Workshop on Contextual Information Access, Seeking and Retrieval Evaluation*, Tolouse, France, 2009.
10. M. Melucci. A basis for information retrieval in context. *ACM Transaction on Information Systems*, 26(3):1–41, 2008.

Maintaining object authenticity in very large digital libraries

Tobias Blanke, Stephen Grace, Mark Hedges, Gareth Knight, and Shrija Rajbhandari

Centre for e-Research, King's College London, UK
tobias.blanke@kcl.ac.uk
stephen.grace@kcl.ac.uk
mark.hedges@kcl.ac.uk
gareth.knight@kcl.ac.uk
shrija.rajbhandari@kcl.ac.uk

Abstract. Digital libraries are increasingly being used to manage research data, leading to a significant increase in the amount of material held in such systems. Much of this material is irreplaceable, and there is a pressing need to maintain long-term access to it; automation of curation is key if a scalable solution is to be found. The concept of significant properties is key to maintaining the integrity and authenticity of a digital object over time and technological change. We present a case study which addresses significant properties as they apply to a working scientific community active in experimental and computational research, and we describe an approach to automating this model by representing curation policies as rules that are implemented using the iRODS middleware.

1 Background

Research across disciplines is increasingly both a generator and user of very large quantities of data, the so-called "data deluge" [5]. This is not an issue only for sciences such as particle physics and astronomy, which have for many years been generating petabyte-scale data sets, but is increasingly the case in subjects such as the humanities, with the growth of digitisation projects producing high-resolution images, video and audio, as well as the existence of born-digital archives. The issue is also not just one of scale; in many disciplines, the information objects created by researchers may be highly complex, with many structural and semantic relationships both internal and contextual. For example, textual scholars may work with a variety of marked-up textual resources, databases and multi-media objects, as well as a number of secondary resources such as dictionaries and concordances; medical researchers may deal with large two-and three-dimensional image files that have detailed annotations and links to other resources. Thus we may speak of a complexity deluge as well as a data deluge. All this raises significant challenges for the curation of the data.

The creation of this digital material represents a considerable investment in intellectual effort, time and, in many cases, public funding. In recent years it has

been increasingly recognised that this investment in research must be protected through long-term management of its outputs. In contrast to the management of physical artefacts, it is considered infeasible to store digital artefacts in their original form and expect them to remain readable and usable when required at a later date. Instead, digital curation is built upon the premise that the environment in which research is accessed and used is likely to change over time and that specific activities, such as format conversion or emulation are required at distinct stages in a digital object's lifecycle to ensure that information remains accessible and usable.

Digital preservation is thus a major issue for research, and indeed for any domain that needs to ensure long-term access to digital material, for example archives, digital libraries and commercial sectors, and much work has gone into developing digital preservation standards such as the OAIS Reference Model [2] and the PREMIS Data Dictionary for preservation metadata [8]. In general, two distinct approaches to digital preservation are discussed; either digital resources are preserved for the future by transforming them into certain standard and normalised formats that one would expect to continue to be comprehensible as the environment changes, or the software environment in which the data is rendered and processed is maintained by enabling its emulation in future software environments [10] [6]. The latter poses significant challenges. Not only will we have to ensure that the code is future-proof and written in a language that will still be understood by future computer systems, we will also have to guarantee that the current code does not contain (fatal) logical errors that might break future systems and that it does not contain harmful code. Not only for these reasons, but also for reasons of scalability, we do not at the moment see this emulation approach as feasible for research data, and in the current work we have concentrated on improving the first option.

2 Significant properties

2.1 Overview

In the work described in this paper, we have followed a normalisation/migration approach; specifically, our approach involves converting a digital object to one of a range of preferred, standard formats at the time of ingest (format normalisation), combined with subsequent conversions of objects throughout their life-cycle as formats or rendering software tools become obsolete. Such technological migrations are not without risk, as it may result in information loss or some change to the way in which information is represented, so the experience of an object may not be identical after conversion. The level of data loss depends on the number and nature of preservation treatments applied to an object, the new data format(s) selected, and the level of human intervention and post-conversion analysis. In order to ensure the authenticity of a digital object throughout its life-cycle, detailed audit information must be captured and retained at each stage, in order to quantify the information loss and thus to provide a measure for the integrity of the data. The definition of the Significant Properties (SPs) of digital

material is key to ensuring that the integrity and authenticity of information is maintained, to enable subsequent access, use and understanding.

Traditional approaches to this audit information (or preservation metadata) have been data-centric, concerned with recording those aspects of the digital object necessary to maintain access to it. However, recent work, by projects such as InSPECT, PLANETS and CASPAR[1], has advocated a sociological methodology that examines the epistemological interpretation of the digital material's creator and its 'designated community', together with the purposes for which it was created and is being used [7]. The challenge for a data curator, therefore is to analyse the requirements of this designated community and identify the characteristics of the digital artefact that enable these requirements to be satisfied. An example of such an approach in given in [3], which presents a preservation system for cultural heritage material that is based on migrating files to preservation formats. After migration, the information loss incurred is determined by applying comparison functions based on significant properties.

These projects have applied an SP approach to a range of generic categories of digital object, from the point of view of data curators. The work described in this paper differs in that we focus on a specific discipline, specific communities of practice within that discipline with their own workflows and objectives, and specific categories of digital object used by these communities, with the aim of investigating how the definition of SPs can contribute to the preservation of these communities' research data.

2.2 Case study

In our case study, we are addressing SPs as they apply to a working scientific community that is active in experimental and computational research and that uses complex digital material on a day-to-day basis. We are not only analysing the data, activities and objectives of this community at a theoretical level, we are also developing a practical demonstration of the application, impact and importance of the significant properties approach to digital curation within this targeted environment.

The discipline is cell and molecular biophysics, an area with a strongly interdisciplinary focus, operating at the interface between the health, biomedical and physical sciences, and which has links to numerous other areas of research within biomedical and health sciences. A variety of digital objects is produced in this research - in the current work we are focusing on information objects that may be broadly described as images. In many cases the raw images are obtained by advanced microscopy or nanoimaging techniques, but these may also include 3D images produced from 'stacks' of 2D images, and time sequences of images capturing temporal development or processes. In other cases the objects are less conventional images generated by detectors specific to particular physical processes, for example diffraction images generated by crystallographic experiments

[1] http://www.significantproperties.org.uk/, http://www.planets-project.eu/, http://www.casparpreserves.eu/

on large molecules such as proteins. Raw objects are frequently in proprietary formats that are dependent on the equipment used to capture them, although the formats are generally open, facilitating the development of software for rendering or processing them. In all cases, these objects are complex in nature and possess properties whose significance is often specific to the domain.

The designated communities are (i) the researchers, who generate and transform digital material in the course of their research activities, and (ii) staff concerned with the management, curation, archiving and preservation of the digital objects generated by the research. The objectives of these two communities are not identical, nor are the workflows in which they involve the digital objects. Researchers are concerned with the rendering of images, and with a variety of transformations of images, in order to facilitate particular research and teaching activities; curation staff are concerned with maintaining the autheniticity and integrity of an object over time. Nevertheless, their objectives are of course strongly connected, as one of the aims of archiving the digital material is to enable subsequent verification of existing research, or future research based on existing experimental data.

As a practical demonstration of our approach, we built services that can be used to (i) extract SPs from an object, (ii) convert an object from one format to another, and (iii) validate the SPs of a converted object against its original profile. These services have been combined to develop a software demonstrator, based on the iRODS system (see below), which implements an SP approach within our targeted area.

As remarked above, any format transformation involves some degree of information loss. Our aim was to identify the properties of the information objects (in our case raster images) that were deemed significant with respect to the future interpretation of those objects by the designated user community, and to express them quantitatively or formally, so that their persistence through images transformations can be verified. Although at a quantitative level there is information loss, this information is not relevant from the point of view of the searchers' visual or processing experience.

There are a number of ways of quantifying information loss. A simple example is to calculate the *root mean square* error between two images. However, as such measures are very generic and give equal weight to all parts of the image, they may not be very helpful in ensuring the persistence of the image characteristics that are of significance in particular contexts. Their limitations become quite clear in the context of image features such as the existence of fine lines, where the rms error measure may manifestly fail to quantify information loss that is immediately evident to visual inspection[2].

Instead of attempting to apply more complex, but still generic, measures of quantifying information, we took the approach of working closely with the researchers that create and use the images, and identifying the characteristics of various categories of image that were of significance to them. The example in the previous paragraph is apposite, as in certain of the images considered

[2] See, for example, [4], p. 536

the existence and location of quite fine lines was of great significance to the researcher, whereas some parts of the image were of little or no relevance.

3 Rule-based automation of preservation policies

To the researchers who creates and use this data, curation activities should be transparent. Data creators want the assurance that their results will persist after the research project is complete, and data users want continued access to the data in a form that remains usable through technological and cultural change. It is the responsibility of archival staff to ensure this persistence. However, as datasets increase in size and complexity, an approach to curation that involves significant manual activity is not sustainable, in particular since specialised knowledge is frequently required to curate data in particular disciplines. Consequently, there is a need to develop approaches that maximise the automation of preservation activities and involve archive staff only when required.

The methodology followed in the work described here is to represent the preservation policies and procedures formally as rules, which specify the sequences of actions that are taken in particular circumstances, or when certain pre-conditions are satisfied. These pre-conditions may include the occurrence of a triggering event, and assertions about the current state of the preservation system or of objects within it. In addition, the rules can incorporate post-conditions, which support verification of any actions that have taken place, for example that the preservation environment is in a consistent state or that authenticity of the preserved objects has been maintained.

These rules are implemented using the iRODS[3] data grid middleware developed by DICE[4] [9]. A particular feature of iRODS is its Rule Engine, which allows data management policies to be represented in terms of rules comprised of pre-defined sequences of actions that are executed in particular circumstances. Rule execution results in the creation of persistent state information, which can be accessed from within rules to track and control subsequent rule execution. These rules can be executed automatically by iRODS as part of its normal execution, in response to certain conditions or triggers. These rules have great potential for implementing data management strategies that are to take place "under the hood", where the data owners need to be confident that certain processing is occurring, but do not want to concern themselves with it.

3.1 Example

Let us now give an example of the use of these rules for implementing digital curation strategies that integrate the idea of maintaining the significant properties of information objects, as described above. Format conversion takes place typically when an information object is ingested into an archive, and on subsequent

[3] http://irods.sdsc.edu
[4] http://dice.unc.edu/

occasions when it is judged that file formats or associated software packages are in danger of obsolescence, putting information content at risk. In this example, we consider the former scenario.

An iRODS rule is defined as follows[5]:

actionDef|*condition*|*workflow-chain*|*recovery-chain*

where *actionDef* is the identifier of the rule, *condition* defines the circumstances under which the rule will be invoked, *workflow-chain* is the sequence of actions that the rule will execute (separated by ##), and *recovery-chain* is the sequence of actions to be executed in case a failure occurs within *workflow-chain* (that is, it defines how a partially executed rule will be rolled back). The actions in a *workflow-chain* can be either atomic actions, know as "micro-services", or rules, thus a rule can be built up cumulatively from other rules.

It may be represented as an iRODS rule as follows:

acPostProcForPut||
acCheckObjectIntegrity##acAnalyseObject##
acNormaliseObject##msiSysReplDataObj(PresRescGrp,all)|
nop##nop##nop##msiCleanUpReplicas

where *acPostProcForPut* is a system action that is executed automatically when an object is put into an iRODS system, and *nop* indicates that no *recovery-chain* component is executed for the corresponding *workflow-chain* component. The components of the *workflow-chain* indicate the various activities that are carried out when the object is ingested, and may be further broken down until the base components are atomic actions or "micro-services", for example:

acNormaliseObject||
acCharacteriseObject##acConvertObject##
acCharacteriseConvertedObject##acValidateConversion|
nop##nop##nop

Here *acCharacteriseObject* and *acCharacteriseConvertedObject*, each of which may comprise several actions, extract the SPs of the original and converted objects, and *acValidateConversion* verifies that the degree of information loss lies within acceptable limits. iRODS also allows conditional execution of rules (see the rule format described above), allowing different rules and conversion services to be configured for different categories of object.

A second feature of iRODS rules that we can exploit is the ability to specify a number of rule definitions corresponding to the same goal, which can be executed in turn (in a preferred order) until one is successful. Unsuccessful rule executions are rolled back by executing the associated *recovery-chain* workflows. To see why this may be useful, consider again the scenario of format conversion. Conversion tools will not perform perfectly in all cases, so in practice it may be necessary to try several before an acceptable result is achieved. Using iRODS rules we can automate this as in the following example, where *msiConvertImage*<*n*> are two micro-services invoking distinct conversion services, and <*imageCategoryA*> is the category of images being addressed:

[5] See http://www.irods.org/index.php/Rules

acNormaliseObject|
$format == <*imageCategoryA*>|
acCharacteriseObject##msiConvertImage1##
acCharacteriseConvertedObject##acValidateConversion|
nop##nop##nop##msiCleanupNormalisation
acNormaliseObject|
$format == <*imageCategoryA*>|
acCharacteriseObject##msiConvertImage2##
acCharacteriseConvertedObject##acValidateConversion|
nop##nop##nop##msiCleanupNormalisation

If an object falls into the appropriate category, the iRODS Rule Engine will invoke the first matching rule. If one of the micro-services within *acValidateConversion* determines that the conversion was unacceptable (e.g. in terms of information loss), it can return an error, in which case the Rule Engine will roll back the rule execution by calling *msiCleanupNormalisation*, and will execute instead the next matching rule in the list[6]. This procedure can be repeated, executing a number of different conversion implementations in turn (in a preferred order) until one is successful. If no rule is successful, curation staff can be notified that manual action is required.

4 Current status and future work

In this paper we have outlined how to implement an iRODS-based data grid system to support digital curation functionality in large archives of potentially complex scientific data. The iRODS Rule Engine allows complex preservation strategies to be written as rules, which can be triggered automatically when certain events occur, for example the ingest of an object into the repository. As these rules can be implemented conditionally, iRODS implements the event-condition-action model known from active database management systems, providing a great degree of flexibility for implementing automated curation applications. In particular, we can use this method to develop concrete implementations of significant property approaches to digital preservation. The initial prototyping demonstrated the feasibility of this approach, and we were encouraged to begin a deeper analysis, developing more extensive sets of rules and applying our approach of identifying significant properties to more categories of research data

Another issue is the difficulty of expressing the significant properties of an information object in a systematic way across different file formats, which may incorporate quite different data structure. While the more generic characteristics of an object can be expressed using standard schemas such as PREMIS [8], this is more difficult with the varied domain- and community-specific characteristics that we are addressing in our current work. However, this is necessary if we are to be able to extract and compare such characteristics automatically. At present,

[6] Note that in a real-life situation each component of *workflow-chain* will have a corresponding *recovery-chain* component; here for clarity all but the last one are omitted.

this is done in a somewhat *ad hoc* way, indeed some knowledge is embedded in the micro-services themselves (i.e. in the software), which is not desirable. In future work, we plan to look at more general ways of describing these, and in particular the XCL (eXtensible Characterisation Language) ontology developed as part of the PLANETS project [1].

References

1. C. Becker, A. Rauber, V. Heydegger, J. Schnasse and M. Thaller. A Generic XML Language for Characterising Objects to Support Digital Preservation. In *SAC08*, Fortaleza, Ceara, Brazil, 2008.
2. Consultative Committee for Space Data Systems. Reference Model for an Open Archival Information System (OAIS). CCSDS, Washington, DC, 2002.
3. M. Ferreira, A.A. Baptista and J.C. Ramalho. An intelligent decision support system for digital preservation. *International Journal on Digital Libraries*, 6(4):295–304, 2007.
4. R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Third Edition, Prentice Hall, 2007.
5. T. Hey and A. Trefethen. The data deluge: an e-Science perspective. In *F. Berman, A. Hey, G. Fox (eds.), Grid Computing: Making the Global Infrastructure a Reality*, John Wiley and Sons, Hoboken, NJ, 2003.
6. J. van der Hoeven, B. Lohman and R. Verdegem. Emulation for digital preservation in practice: the results. *Int. J. of Digital Curation*, 2(2), 2007.
7. G. Knight. Framework for the definition of significant properties. Project report, Arts and Humanities Data Service, London, UK, 2008.
8. PREMIS Data Dictionary for Preservation Metadata. Version 2.0, PREMIS Editorial Committee, Washington DC, 2008.
9. A. Rajasekar, M. Wan, R. Moore, W. Schroeder. A Prototype Rule-based Distributed Data Management System. In *HPDC workshop on Next Generation Distributed Data Management*, Paris, France, 2006.
10. K. Thibodeau. Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years. In *Proceedings of The State of Digital Preservation: An International Perspective*, Washington DC, USA, 2002.